

# SISTEM PENJADWALAN AIR CONDITIONER (AC) RUANGAN BERDASARKAN JADWAL MATAKULIAH MENGGUNAKAN ESP8266, PIR SENSOR DAN ANDROID

*(Air Conditioner Scheduler based on courses schedule using ESP8266, PIR Sensor and Android)*

Muhammad Malik Saputra, I Gede Putu Wirarama W.W.\* , Ariyan Zubaidi

Dept Informatics Engineering, Mataram University  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: jackmalik79@gmail.com, [wirarama, zubaidi13]@unram.ac.id

## Abstract

*In the scope of the University of Mataram (UNRAM) Informatics Engineering Study Program (PSTI), currently has a room temperature controller or commonly called an Air Conditioner (AC) to support teaching and learning activities but often its use is not organized such as leaving the air conditioner on even after the classroom empty or no activities that cause inefficient use of the air conditioner which can also cause damage to the air conditioner. Therefore, it is necessary to have a system that controls the AC scheduling automatically which is more efficient. This can be realized with the Internet of Things (IoT) technology. So it is necessary to build a system that will make the AC automatically turn on according to the class schedule using a microcontroller (ESP8266 Wemos D1) with an infrared transmitter to control AC and motion detection which will use the MQTT protocol as a broker. Passive Infrared (PIR) motion sensor The sensor will detect movement in the classroom which becomes a condition for turning off the air conditioner if after 30 minutes there is no movement in the room. For operation using an Android smartphone as well as an interface for entering class schedules, only PSTI UNRAM staff will operate the application. That way the air conditioner can automatically turn on according to the course schedule and can turn off automatically after 30 minutes if there is no person in the classroom.*

**Keywords:** IOT, PIR Sensor, Android, ESP8266, WEMOS, MQTT, AC

\*Penulis Korespondensi

## 1. PENDAHULUAN

*Air Conditioner* atau lebih dikenal dengan nama AC merupakan suatu peralatan yang berfungsi untuk mengkondisikan suhu atau temperatur udara dalam suatu ruangan atau dengan kata lain untuk mengkondisikan suhu udara atau temperatur udara dalam suatu ruangan yang disesuaikan dengan kondisi tubuh penghuni ruangan tersebut[1].

Ruang kuliah yang baik sangat penting untuk kelangsungan kegiatan belajar mengajar. Pada ruang lingkup Program Studi Teknik Informatika (PSTI) Universitas Mataram (UNRAM) saat ini memiliki pengatur suhu ruangan atau biasa disebut dengan *Air Conditioner*(AC) untuk menunjang kegiatan belajar mengajar. Namun, pengoperasian AC juga perlu diperhatikan, karena tidak jarang pengoperasian AC pada ruang kuliah tidak efisien seperti membiarkan AC tetap menyala bahkan setelah ruang kelas kosong atau tidak ada kegiatan yang menyebabkan .

Berdasarkan tanya jawab yang telah dilakukan dengan staf PSTI UNRAM, pengoperasian AC masih dilakukan secara manual yaitu dengan membawa *remote control* pada saat jam perkuliahan berlangsung, dan juga telah dilakukan survey terhadap beberapa mahasiswa yang seringkali menemukan AC masih tetap menyala saat ruang kelas kosong bahkan sampai keesokan harinya karna kelalaian pengguna. Adapun kasus AC menyala bukan hanya dari penggunaan saat jam matakuliah berlangsung, seperti mahasiswa yang iseng menyalakan AC dengan aplikasi tertentu dan membiarkannya menyala begitu saja.

*Internet Of Things* atau yang biasa dikenal dengan IoT merupakan bentuk koneksi suatu perangkat yang saling terhubung dan mampu menghasilkan suatu informasi yang dapat diakses dan digunakan oleh manusia atau sistem[2].

*Protokol Message Queue Telemetry Transport* (MQTT) adalah protokol pesan yang sangat sederhana dan ringan. Protokol MQTT menggunakan arsitektur

*publish/subscribe* yang dirancang secara terbuka dan mudah untuk diimplementasikan, yang mampu menangani ribuan *client* jarak jauh dengan hanya satu server. MQTT meminimalkan bandwidth jaringan dan kebutuhan sumber daya perangkat ketika mencoba untuk menjamin kehandalan dan pengiriman. Pendekatan ini membuat protokol MQTT sangat cocok untuk menghubungkan mesin ke mesin (M2M), merupakan aspek penting dari konsep *Internet of things*[4].

Berdasarkan uraian permasalahan yang dihadapi maka akan dibangun sistem guna membuat AC otomatis menyala sesuai jadwal kuliah menggunakan mikrokontroler dengan *infrared transmitter* untuk mengendalikan AC dan deteksi gerak yang akan menggunakan protokol MQTT sebagai broker. Sensor gerak yang digunakan adalah *Passive Infrared Sensor* (PIR), untuk mendeteksi gerakan pada ruang kelas yang akan menjadi kondisi untuk mematikan AC jika setelah 30 menit tidak ada pergerakan didalam ruangan. Pada penelitian ini digunakan *smartphone(mobile apps)* yaitu aplikasi android.

## 2. TINJAUAN PUSTAKA

Penelitian yang sudah dilakukan sebelumnya adalah penelitian[6], dimana perangkat IoT ini dibuat menggunakan sensor PIR untuk mendeteksi pergerakan manusia dan DHT22 untuk mendeteksi suhu dan kelembaban udara, serta protokol MQTT untuk komunikasi data antara perangkat IoT dengan sistem web. Berdasarkan skenario pengujian ruangan berisi mahasiswa dan ruang kosong yang telah dilakukan, perangkat IoT yang dibuat dapat melakukan manajemen penggunaan AC secara otomatis.

Namun Sistem ini berfokus pada pergerakan dan suhu, jadi pengoperasian ACnya akan dilakukan pada saat gerakan terdeteksi saja, yang dikombinasikan dengan DHT22 untuk mengukur kelembaban suhu yang kan didapatkan.

Pada penelitian[7], telah dilakukan rancangan-bangun sistem otomatisasi kontrol lampu berdasarkan keberadaan orang didalam ruangan. Sistem kontrol lampu akan menyala selama ada orang didalam ruangan, dan lampu akan mati ketika orang meninggalkan ruangan. Keberadaan orang akan di deteksi oleh sensor *passive infrared* (PIR). *Relay* digunakankan untuk menghubungkan antara arus DC dan arus AC[7].

Pada penelitian[7], terdapat kesamaan berupa penggunaan sensor PIR untuk mengetahui pergerakan yang ada pada ruangan tersebut guna menghidupkan lampu saat ada pergerakan di dalam ruangan atau

selama ada orang yang berada didalam ruangan tersebut. namun penelitian ini masih ada kekurangan yaitu membutuhkan waktu untuk pemanasan selama 25,52 detik.

Adapun penelitian[8] Telah dilakukan penelitian terhadap penggunaan AC pada ruang kelas di Universitas Serang Raya yang beroperasi tidak sesuai dengan kebutuhan sehingga hal ini mengakibatkan pemborosan energi. Tujuan dari penelitian dan pembuatan alat ini adalah untuk merancang suatu alat yang mampu memantau suhu dan juga kelembaban udara ruangan yang berbasis IoT (*Internet Of Things*) dan dapat menjadi sistem kendali otomatis terhadap suhu ruangan untuk bisa memonitor suhu ruang kelas dari mana saja dan dapat menghemat biaya tagihan listrik. [8].

Pada penelitian[8], menggunakan objek yang sama berupa AC yang akan dioperasikan melalui konsep IoT. Adapun kelebihan dari penelitian ini dapat memonitor suhu kelembaban status AC dan jumlah orang yang berada di ruangan kelas yang mana informasi tersebut dapat menjadi acuan untuk menghidupkan atau mematikan AC.

Penelitian lain yang serupa[9], berjudul Perancangan Alat Monitoring *Air Conditioner* Menggunakan Mikrokontroler Wemos. Untuk menjaga kenyamanan termal dalam ruangan dibutuhkan sebuah alat monitoring *air conditioner*. Alat monitoring dibutuhkan untuk mengukur beberapa parameter penting yang mempengaruhi kondisi termal, diantaranya temperatur, kelembaban, status *power air conditioner*, keberadaan manusia, serta pencatat waktu *real time*. Maka dari itu, pemecahan masalah ini bertujuan untuk merancang sebuah alat monitoring *air conditioner* dengan menggunakan mikrokontroler Wemos. Hasil pengujian digunakan untuk mengetahui kinerja dari setiap modul yang digunakan untuk mengukur parameter yang ada[9].

Dari penelitian[9], diketahui penelitian ini menggunakan mikrokontroler yang sama dengan yang penulis gunakan yaitu ESP-8266, dengan menggunakan Wemos, biaya yang dikeluarkan untuk membangun sistem Wi-Fi dapat terealisasi. Namun pada penelitian Perancangan Alat Monitoring *Air Conditioner* Menggunakan Mikrokontroler Wemos ini hanya melakukan monitoring saja, tidak melakukan perubahan yg signifikan pada AC seperti menghidupka atau mematikan, dan juga tidak menggunakan software tambahan seperti web atau android sehingga kurang efektif jika hanya dimonitor di dalam ruangan saja.

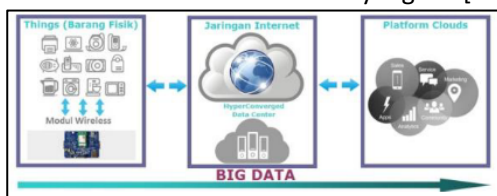
Penelitian yang serupa[10], berjudul Implementasi Kendali Otomatis Lampu dan AC(Air Conditioning). Implementasi Kendali Otomatis Lampu dan AC ini menggunakan konsep *Internet of Things* sebagai arsitektur, karena dengan konsep inilah penjadwalan pada ruangan atau kelas-kelas bisa digunakan dengan kebutuhan yang semestinya. Hasil dari penelitian implementasi kendali otomatis lampu dan AC yaitu pengguna didalam gedung atau ruang dapat mengontrol sistem kendali lampu dan AC dengan mengakses *smartphone* melalui jaringan lokal ataupun internet. *Framework* OpenHAB dapat berjalan dengan baik pada server *smart home system*[10].

Pada penelitian[10], menggunakan mikrokontroler wemos yang terhubung dengan kebutuhan untuk fungsi AC. Pada penelitian ini juga menggunakan *smartphone* sebagai alat kendalinya yang dapat bekerja berdasarkan penggunaan kebutuhan fungsional pada saat penjadwalan atau kebutuhan lainnya. Namun otomatisasi yang dilakukan terhadap AC hanya bergantung pada PIR sensornya saja sehingga tidak tepat untuk digunakan pada ruang kelas.

### 2.1 Internet Of Things (IoT)

*Internet Of Things* (IoT) merupakan bentuk koneksi suatu perangkat yang saling terhubung dan mampu menghasilkan suatu informasi yang dapat diakses dan digunakan oleh manusia atau sistem lainnya. Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 dimana benda-benda disekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan seperti internet. Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. [11].

Setiap objek/ *things* mampu diidentifikasi melalui sistem komputasi yang tertanam dan mampu beroperasi dalam infrastruktur internet yang ada[12].



Gambar 1. Konsep IoT[13]

### 2.2 Android

Android merupakan salah satu sistem operasi *mobile* yang tumbuh di tengah sistem operasi lain yang berkembang saat ini. Android adalah platform komprehensif bersifat *open source* yang dirancang untuk perangkat *mobile*. Android adalah sistem operasi

berbasis Linux yang diperuntukkan untuk *mobile device*. Android merupakan sistem operasi yang paling diminati di masyarakat karena memiliki kelebihan seperti sifat *open source* yang memberikan kebebasan para pengembang untuk menciptakan aplikasi[14].

Untuk melakukan penyimpanan data pada *database*, sistem Android menggunakan SQLite yang merupakan suatu *open source database* yang cukup stabil dan banyak digunakan pada banyak *device* berukuran kecil[15].

### 2.3 Wemos

Wemos adalah sebuah mikrokontroler pengembangan berbasis modul mikrokontroler ESP-8266. Wemos dibuat sebagai solusi dari mahalnya sebuah sistem *wireless* berbasis mikrokontroler lainnya. Dengan menggunakan Wemos, biaya yang dikeluarkan untuk membangun sistem Wi-Fi lebih murah dan kemampuannya untuk menyediakan fasilitas konektivitas Wi-Fi dengan mudah serta memori yang digunakan sangat besar yaitu 4 MB. [9].

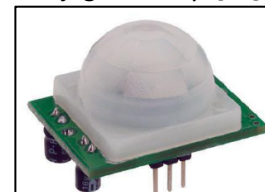


Gambar 2. Bentuk fisik dari Wemos D1[17].

Bahasa pemrograman sama seperti arduino sehingga fungsi setup pin dan fungsi *loop* sama.

### 2.4 Sensor Passive Infrared(PIR)

Sensor PIR (*Passive Infrared*) adalah suatu alat yang berfungsi untuk mengindra atau menangkap suatu besaran fisis (temperatur suhu tubuh manusia) dan merubahnya kebentuk sinyal listrik. Sesuai namanya, *Passive Infrared*, sensor ini bersifat pasif. Sensor ini menerima sinyal *infrared* yang dipancarkan oleh suatu objek yang bergerak (dalam hal ini tubuh manusia). Saat ini dipasaran banyak sekali terdapat jenis sensor PIR, seperti halnya peralatan elektronik yang lainnya, harganya tergantung dari negara pembuat, kualitas dan juga merknya[18].



Gambar 3. Bentuk fisik dari PIR sensor[6].

Sensor PIR mempunyai dua elemen sensing yang terhubung dengan masukan, seperti gambar 2.5. Jika ada sumber panas yang lewat di depan sensor

tersebut, maka sensor akan mengaktifkan sel pertama dan sel kedua sehingga akan menghasilkan bentuk gelombang seperti ditunjukkan dalam gambar 2.3[18]

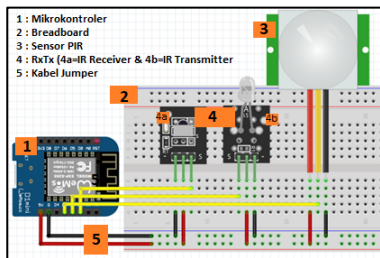
### 3. METODE PENELITIAN

#### 3.1 Rencana Pelaksanaan

Adapun rencana pelaksanaan penelitian sistem penjadwalan AC ruangan berdasarkan matakuliah secara otomatis dapat dilihat pada tahap-tahap seperti studi literatur, analisis kebutuhan sistem, perancangan perangkat keras dan lunak, implementasi serta pengujian dan evaluasi beserta dokumentasi.

#### 3.2 Rancangan Perangkat Keras

Gambaran untuk rancangan perangkat keras dapat dilihat pada Gambar 3.3. Terdapat Wemos D1 yang terhubung ke perangkat RxTx(4) dan Sensor PIR(3) melalui breadboard(2) dengan kabel jumper(5) yang terhubung ke masing-masing perangkat. Wemos D1(1) sebagai mikrokontroler yang mengontrol dan mengolah data yang diterima IR Receiver(4a) serta mengirim data melalui IR Transmitter(4b). Setiap perangkat berfungsi sesuai jenis perangkatnya yaitu IR Receiver(4a) untuk mendeteksi raw data dari remot AC yang digunakan, IR Transmitter(4b) untuk mengirimkan perintah ke AC melalui infrared dan PIR sensor untuk mendeteksi adanya gerakan pada ruangan.



Gambar 4. Rancangan Perangkat Keras

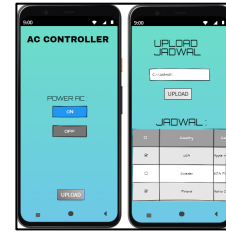
#### 3.3 Rancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak, dilakukan perancangan sistem untuk merancang aplikasi Android juga akan dilakukan perancangan komunikasi MQTT.

##### 3.3.1 Desain Sistem

Berikut merupakan rancangan dari desain sistem yang akan dibuat :

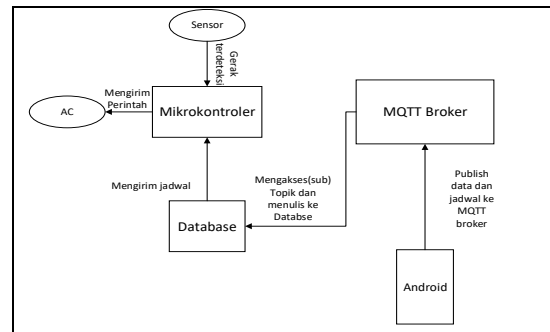
Pada Gambar 3.6 merupakan desain beranda untuk antar muka sistem yang akan dibuat. Pada halaman beranda terdapat toggle untuk menghidupkan dan mematikan AC, kemudian ada slider untuk mengatur suhu, juga terdapat tombol untuk meng-upload jadwal dan melihat jadwal.



Gambar 5. Rancangan beranda dan halaman upload jadwal aplikasi android

##### 3.3.2 Rancangan Komunikasi MQTT

Pada Gambar 3.7 adalah rancangan komunikasi MQTT dari sistem yang akan dibuat. Alur proses dimulai dari mikrokontroler yang melakukan pembacaan data gerak dari sensor. Mikrokontroler mengambil jadwal penggunaan AC dari database. Aplikasi android yang dapat melakukan publish melalui MQTT broker dan mengirimkannya database yang akan tersimpan dan kemudian diteruskan ke mikrokontroler.



Gambar 6. Rancangan Komunikasi MQTT

#### 3.4 Implementasi Sistem

Pada tahap implementasi, akan dilakukan penyusunan konfigurasi perangkat keras dan pembangunan perangkat lunak dari rancangan yang telah dibuat sebelumnya serta menghubungkan keduanya menjadi satu sistem dengan menggunakan protokol MQTT sebagai media pengiriman data.

Jika sistem sudah sesuai dengan kebutuhan maka akan dilanjutkan menuju tahap dokumentasi dan laporan. Sedangkan jika sistem belum berjalan sesuai dengan kebutuhan maka akan dilakukan perbaikan mulai dari tahap perancangan perangkat keras.

#### 3.5 Dokumentasi dan Laporan

Setelah melakukan pengujian sistem dan berjalan dengan semestinya, maka selanjutnya akan didokumentasikan dan diambil kesimpulan berdasarkan dokumen tersebut. Kemudian dari kesimpulan yang telah didapatkan selanjutnya akan digunakan sebagai acuan untuk pengembangan sistem selanjutnya.

**3.6 Rencana Pelaksanaan**

Estimasi waktu yang digunakan dalam proses pengembangan sistem pada penelitian ini yaitu selama kurang lebih 5 minggu yang dimulai dari bulan juni 2020.

**3.7 Realisasi Sistem**

Bab ini akan membahas tentang hasil dari penelitian yang dilakukan sesuai dengan judul penelitian yaitu “Sistem Penjadwalan AC Ruang Berdasarkan Jadwal Matakuliah Menggunakan ESP8266, PIR Sensor, dan Android”. Bentuk Realisasi yang akan dilakukan telah dibuat berdasarkan rancangan yang telah dijelaskan pada bab sebelumnya. Pada bab ini juga akan dijelaskan tentang pengujian pada keseluruhan sistem serta mengevaluasi sistem yang berjalan agar sesuai dengan yang telah direncanakan.

**4. HASIL DAN PEMBAHASAN**

**4.1. Realisasi Penyusunan Perangkat Keras**

Realisasi penyusunan perangkat keras mengacu pada rancangan yang terdapat pada bab sebelumnya. Realisasi perangkat keras yang telah dibuat dapat dilihat pada Gambar 4.1 di bawah ini.

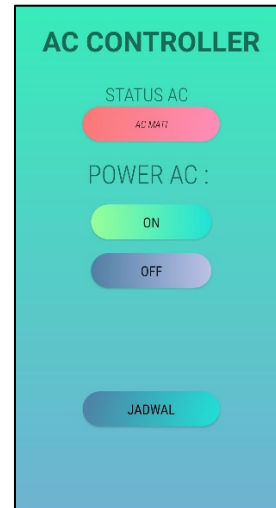


Gambar 7. Realisasi Perangkat Keras.

Pada Gambar 7, terdapat lima perangkat keras yang dihubungkan menjadi sebuah perangkat untuk melakukan manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM, yang terdiri dari Wemos D1, IR Receiver, IR Transmitter, dan sensor PIR.

**4.2. Realisasi Pembangunan Aplikasi Android**

Realisasi pembangunan aplikasi android sistem manajemen penggunaan AC secara otomatis dibuat sesuai dengan perancangan pada bab sebelumnya dengan beberapa penyesuaian. Bahasa yang digunakan dalam pembangunan aplikasi ini adalah java dan menggunakan Android Studio sebagai IDE. Berikut adalah tampilan dari aplikasi android yang telah dibuat. Tampilan *Home*:



Gambar 8. Tampilan menu awal aplikasi

Gambar 8 diatas merupakan tampilan beranda dari aplikasi android. Terdapat 3 tombol yang memiliki fungsi berbeda. Tombol “ON” berfungsi untuk mengirim data “1” ke broker MQTT untuk diambil oleh *subscriber* yaitu Wemos D1 dan akan menghidupkan AC. Tombol “OFF” berfungsi untuk mengirim data “0” ke broker MQTT untuk diambil oleh *subscriber* yaitu Wemos D1 dan akan mematikan AC. Sedangkan tombol “JADWAL” berfungsi untuk berpindah ke halaman jadwal. Dan ada indikator status AC pada bagian paling atas untuk mengetahui AC sedang menyala atau tidak.

Tampilan Menu Jadwal :



Gambar 9. Tampilan Menu Jadwal

Gambar 9 diatas merupakan tampilan menu jadwal dari aplikasi android. Terdapat beberapa kolom yang menampilkan nama matakuliah yang tentunya berfungsi untuk menampilkan bagian dari jadwal yang sudah ada dalam *database*. Sekaligus menjadi *Primary Key*.

Tampilan *Form* Tambah dan Edit Jadwal:

Gambar 10. Tampilan Menu *Form* Tambah dan Edit Jadwal Aplikasi

Gambar 10 diatas merupakan tampilan edit dan tambah jadwal dari aplikasi android. Terdapat beberapa kolom yang memiliki fungsi berbeda. Diantaranya adalah kolom Ruang untuk memilih ruang kelas, kolom hari untuk menentukan hari, kolom nama mata kuliah, kolom jam untuk menentukan waktu mulainya matakuliah yang akan diinput dan ada kolom dosen untuk meng-*input* kode dosen yang mengampu mata kuliah tersebut. Tombol “KIRIM” disini berfungsi untuk mengirimkan data-data yang telah diinput ke broker yang kemudian akan diambil oleh *script* yang berjalan di komputer untuk melakukan *input* kedalam database MySQL.

#### 4.3. Realisasi Pembangunan Komunikasi Data

Pada realisasi pembangunan arsitektur komunikasi data MQTT menggunakan *script* Python yang akan berjalan pada laptop. *Script* ini akan mengambil data yang dikirimkan melalui android dan meng-*input*nya ke *database*, atau mengirimnya langsung ke mikrokontroler. *Script* ini juga mengambil data yang ada pada *database* lalu memutuskan mengirimkan MQTT untuk menghidupkan atau mematikan AC. Mikrokontroler hanya menunggu perintah dari laptop untuk hidup mati AC saja, namun laptop juga tidak berhubungan langsung dengan *user*, *user* hanya berhubungan langsung dengan android sehingga laptop dan mikrokontroler Wemos murni komunikasi mesin ke mesin. Berikut penggalan *Script* Python yang digunakan.

```

1. def on_connecta(mqtta, obj, flags, rc):
2.     print("rc: " + str(rc))
3. def on_subscribea(mqtta, obj, mid, granted_qos):
4.     print("Subscribed: " + str(obj))
5. mqtta.connect("broker.hivemq.com", 1883)
6. mqtta.subscribe(["jmcdh",1], ("jmcd000",1), ("jmcd001",1), ("jmcd002",1)])

```

Kode Sumber 1 *Script* deklarasi fungsi koneksi dan *subscribe*

*Script* diatas merupakan pembentukan fungsi koneksi MQTT dan fungsi *subscribe*. Untuk melakukan koneksi ke broker MQTT menggunakan fungsi “on\_connecta” yang telah dipanggil pada *script* “mqtta.connect(“ broker.hivemq.com”, 1883 )” bermaksud melakukan koneksi kepada “broker.hivemq.com” pada port “1883”. Kemudian untuk melakukan *subscribe* menggunakan fungsi “on\_subscribea” yang telah dipanggil untuk mensubscribe beberapa topik yang ada pada *script* “mqtta.subscribe”.

```

1. def on_messagea(mqtta, obj, msg):
2.     on_har(msg)
3.     on_rua(msg)
4.     on_nama(msg)
5.     on_ja(msg)
6.     on_namd(msg)
7.     if hari and ruang and namat and jam and namdos != None:
8. mycursor = mydb.cursor()
9. sql = "INSERT INTO jadwal (Hari,Ruangan,Namat,Jam,Namdos) VALUES (%s,%s,%s,%s,%s) ON DUPLICATE KEY UPDATE Namat=%s"
10. val = (hari,ruang,namat,jam,namdos,namat)
11. mycursor.execute(sql,val)
12. mydb.commit()
13. print(mycursor.rowcount, "record inserted.")

```

Kode Sumber 2 *Script* deklarasi fungsi *input* data ke dalam *database*

*Script* diatas merupakan pembentukan fungsi untuk meng-*input* pesan yang telah dikirim melalui broker untuk disimpan kedalam *database* MySQL. Dalam fungsi tersebut terdapat pemanggilan fungsi lain yaitu “on\_har(msg)” dkk. yang bertujuan untuk mengambil pesan dari topik yang berbeda-beda-beda. Setelah pesannya didapatkan akan melakukan cek kondisi apakah variabel dari setiap fungsi tidak bernilai kosong, jika benar



maka *sql query* akan dijalankan dan melakukan *input* ke *database*. Adapun *script* yang membentuk fungsi-fungsi untuk mengambil pesan setiap topik sebagai berikut:

```
1. def getpub():b=x.strftime("%m")
1. d=x.strftime("%d")
2. mycurso = mydb.cursor()
3. sql = "SELECT * FROM log ORDER BY
`Log Waktu` DESC LIMIT 1"
4. mycurso.execute(sql)
5. logwa, logsa=mycurso.fetchone()
6. lwj=logwa.strftime("%H")
7. lwh=logwa.strftime("%d")
8. lwb=logwa.strftime("%m")
```

Kode Sumber 3 Deklarasi variabel lanjutan

*Script* diatas merupakan pendeklarasian variabel lanjutan dari *script* sebelumnya, dimana *Script* "b=x.strftime("%m")" untuk membuat variabel "b" sebagai wadah untuk bulan untuk range 1-12, sebagaimana variabel "b", variabel "d" berfungsi menampung tanggal. Kemudian dijalankan *syntax query*"sql = "SELECT \* FROM log ORDER BY `Log Waktu` DESC LIMIT 1" untuk mengambil laporan pada *database* setelah itu akan dilakukan kondisi untuk mengetahui jika sekarang adalah hari dan jam yang sama dengan data yang sudah tersimpan pada *database* dengan *script* berikut :

```
1. if y==day1 and z==jam1 and v>=menit1:
2. sql=("SELECT Hari,Jam FROM jadwal
where Hari=%s AND Jam=%s ")
3. val=("Senin",7)
4. mycurso.execute(sql,val)
5. mycurso.fetchone()
6. print(mycurso.rowcount)
7. if mycurso.rowcount==1:
8. if lwj==z and lwb==b and lwh==d
and logsa=="Hidup":
9. print("AC Sedang Menyala")
10. else:
11. v =1
12. publish.single("onindong", v,
hostname="broker.hivemq.com")
13. e="Hidup"
14. sql = "INSERT INTO log(`Log Waktu`,`Log Status`)VALUES(%s,%s)"
15. val = (t,e)
16. mycurso.execute(sql,val)
17. mydb.commit()
18. print(mycurso.rowcount, "Data Diterima, AC Telah Dihidupkan.")
19. else:
20. print("Jam Kosong")
```

Kode Sumber 4 *Script* kondisi untuk mengirimkan pesan MQTT sesuai jadwal

*Script* diatas merupakan kondisi yang akan menyesuaikan waktu saat ini dengan yang ada pada jadwal di *database* menggunakan dua kondisi. Kondisi yang pertama akan melakukan pengecekan jika hari, jam dan menit saat ini sama dengan yang ada di variabel atau tidak, jika benar maka akan melakukan proses pengambilan data hari dan jam pada *database* yang kemudian akan dilakukan pengecekan data hari dan jam yang diambil dari *database* tadi, jika data yang diambil tidak kosong. Setelah itu akan berlanjut ke kondisi berikutnya untuk memeriksa apakah sudah pernah dilakukan pengiriman MQTT yang ditandai dengan telah di-*uploadnya* laporan log waktu (berisi tanggal dan jam) dan log status(berisi data "hidup" dan "mati") ke *database*. Jika kondisi pertama tidak terpenuhi yaitu jika hari, jam dan menit sama maka *script* tidak akan melakukan proses apapun dan menampilkan text "Belum Ada Kelas".

```
1. elif y==day1 and z==jamis and v>=menitis:
2. v =0
3. publish.single("offindong", v, hostname="broker.hivemq.com")
4. mycursorr = mydb.cursor()
5. e="Mati"
6. sql = "INSERT INTO log(`Log Waktu`,`Log Status`)VALUES(%s,%s)"
7. val = (t,e)
8. mycursorr.execute(sql, val)
9. mydb.commit()
10. print(mycursorr.rowcount, "record inserted.")
11. print("Jam Istirahat, AC Dimatikan")
```

Kode Sumber 5 Kondisi untuk mengontrol AC dan meng-*input* laporan

Adapun kondisi AC dimatikan ada dua kondisi yaitu saat jam istirahat dan jam pulang. *Script* diatas adalah untuk mematikan AC saat jam istirahat dimana jika hari,jam dan menit sama dengan saat ini maka akan menghidupkan AC sesuai perintah, sama halnya dengan waktu jam pulang menggunakan *script* seperti di atas namun berbeda kondisi awal yang semula "jamis" dan "menitis" untuk jam istirahat diganti menggunakan "jampul" dan "menitpul" untuk jam pulang. Begitupun dengan kondisi hari berikutnya sampai hari terakhir "day5" atau hari jumat.

#### 4.4. Realisasi Pembangunan Program Pada Mikrokontroler

Realisasi program pada mikrokontroler dilakukan dengan aplikasi Arduino IDE. Adapun program yang dibangun pada mikrokontroler berupa koneksi internet dan MQTT untuk menerima pesan dari broker yang kemudian mengirimkan kode untuk menghidupkan atau mematikan AC sesuai pesan yang diterima dari broker ataupun PIR sensor. Juga fungsi IR Receiver untuk merekam data remot AC yang digunakan di ruangan A3-02.

```
1. const uint16_t kIrLed = D2;
2. IRsend irsend(kIrLed);
3. uint16_t ondong[139] = {8986, ...,
  650};
4. uint16_t offdong[139] =
  {8982, ..., 652};
```

Kode Sumber 6 Deklarasi variabel mikrokontroler

Script diatas merupakan deklarasi variabel dari library yang digunakan dan deklarasi variabel tambahan pada program. Adapun script "uint16\_t ondong[139]" dan "uint16\_t offdong[139]" untuk mendeklarasikan RAW data yang telah diambil dengan IR receiver dan remote AC.

```
1. void callback(char* topic, byte*
  payload, unsigned int length) {
2.   Serial.print("Message arrived
  ");
3.   Serial.print(topic);
4.   Serial.print("] ");
5.   for (int i = 0; i < length;
  i++){
6.
7.     Serial.print((char)payload[i]);
8.     Serial.println();
9.     if ((char)payload[0] == '1') {
10.      irsend.sendRaw(ondong, 59, 38);
11.      digitalWrite(BUILTIN_LED, LOW);
12.    }
13.    else {
14.      irsend.sendRaw(offdong, 59,
  38);
15.      digitalWrite(BUILTIN_LED,
  HIGH);
16.      stat=0;}}
```

Kode Sumber 7 Fungsi subscribe pada program di mikrokontroler

Script diatas merupakan fungsi subscribe yang juga bisa berperan sebagai publisher atau biasa disebut dengan nama callback. Fungsi ini melakukan subscribe terhadap dua topik yaitu "ondong" untuk menerima pesan "1" yang berfungsi menghidupkan AC dan topik

"offdong" untuk menerima pesan "0" yang berfungsi mematikan AC. Terdapat kondisi dimana jika program menerima pesan berisi "1" maka akan menjalankan script "irsend.sendRaw(ondong, 59, 38)" yang berfungsi untuk menghidupkan AC, begitupun sebaliknya, jika program menerima pesan berisi "0" maka akan menjalankan script "irsend.sendRaw(offdong, 59, 38)" yang berfungsi untuk mematikan AC.

```
1. void loop() {
2.   irsend.begin();
3.   #if ESP8266
4.     Serial.begin(115200, SERIAL_8N1,
  SERIAL_TX_ONLY);
5.   #else // ESP8266
6.     Serial.begin(115200,
  SERIAL_8N1);
7.   #endif // ESP8266
8.   if (!client.connected()) {
9.     reconnect(); }
10.  client.loop();
11.  val=digitalRead(pirPin);
12.  unsigned long now = millis();
13.  if (stat==1 and now -
  lastMsg > 1800000 and val==LOW) {
14.    lastMsg = now;
15.    irsend.sendRaw(offdong, 59,
  38);
16.    stat=0;
17.    digitalWrite(BUILTIN_LED,
  HIGH);
18.  }
19.  else if(now - lastMsg >
  1800000){
20.    lastMsg = now;}
21.  }
```

Kode Sumber 8 Fungsi loop pada program di mikrokontroler

Script diatas merupakan fungsi loop yang akan berjalan terus menerus saat mikrokontroler sudah diaktifkan dan menjalankan fungsi setup. Selama melakukan perulangan, fungsi akan selalau melakukan pemeriksaan koneksi, jika tidak terhubung ke jaringan wifi, maka akan memanggil fungsi "reconnect()" setelah itu melakukan koneksi ke broker dengan script "client.loop()". Pada fungsi loop ini terdapat kondisi untuk mematikan AC dengan bantuan PIR sensor, berdasarkan parameter dari script "if (stat==1 and now - lastMsg > 1800000 and val==LOW) {" fungsi ini akan mematikan AC jika "stat==1" yang berarti AC sedang menyala, dan "now - lastMsg > 1800000 and val==LOW" berarti selama 30 menit tidak ada pergerakan mahluk hidup di ruangan, maka AC akan dimatikan dengan script "irsend.sendRaw(offdong, 59,



38);” kemudian membuat status AC menjadi mati “stat=0”.

**4.5. Pengujian dan Evaluasi Sistem**

Pengujian sistem adalah proses pengujian perangkat keras dan perangkat lunak sistem untuk mengetahui apakah sistem sudah berjalan sesuai dengan yang diinginkan peneliti. Kemudian proses evaluasi sistem dilakukan jika sistem belum berjalan sesuai dengan yang diharapkan peneliti. Pengujian yang dilakukan terhadap sistem sesuai dengan yang perangkat yang digunakan termasuk perangkat keras dan perangkat lunak dan juga fungsi keseluruhan sistem . Adapun metode pengujian yang digunakan adalah metode Black Box Testing dan dilakukan di ruang kelas A3-02 PSTI UNRAM.

**4.6. Hasil Pengujian Sistem Aplikasi Android**

Pengujian untuk aplikasi android dilakukan dengan menggunakan metode *black box testing* untuk menguji fungsi dari fitur aplikasi android berjalan seperti yang diinginkan. Adapun hasil pengujian yang telah dilakukan .

TABEL I. HASIL PENGUJIAN APLIKASI ANDROID

No	Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Tombol ON atau OFF ditekan	Mengirim pesan MQTT 1 untuk tombol ON dan 0 untuk tombol OFF	SESUAI
2	Tombol kirim ditekan	Merngirim data yang ada pada form yang sudah diisi, dengan MQTT.	SESUAI
3	Melakuka n kirim data yang sudah ada	Data tidak akan tersimpan di <i>database</i>	SESUAI
4	Menekan Tombol ON/OFF	Jika Mikrokontroler menyala akan menghidupkan/ matikan indikator	SESUAI
5	Menekan tombol indikator	Tidak merubah status AC	SESUAI
6	Menekan ON/OFF saat mati lampu	Tidak merubah status AC	SESUAI

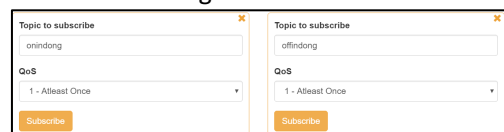
7	Menekan ON/OFF saat tidak ada internet	Tidak merubah status AC	SESUAI
---	--	-------------------------	--------

Pengujian dilakukan dengan bantuan aplikasi MQTTBox untuk menerima pesan MQTT yang dikirim dari android ke broker dengan topik tertentu. Berikut tampilan awal dari MQTTBox.

Adapun hasil dari pengujian aplikasi android dengan menggunakan topik yang sudah dijelaskan sebelumnya adalah sebagai berikut:

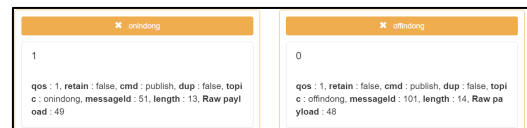
1. Fungsi ON OFF

Untuk fungsi ON OFF digunakan topik “onindong” untuk ON dan “offindong” untuk OFF.



Gambar 9. *Subscribe* untuk tombol ON dan OFF

Pada Gambar 9 dilakukan penulisan topik beserta QoS(Quality Of Service) yang digunakan pada MQTTBox , dan setelah tomol subscribe dipencet maka MQTTBox siap menerima pesan dari Android melalui broker.

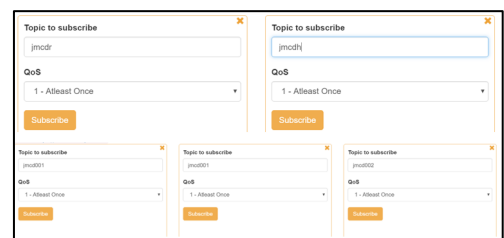


Gambar 10. Hasil pesan yang diterima *subscribe* ON dan OFF

Pada Gambar 4.7 Pesan telah diterima oleh kedua topik dimana untuk fungsi ON mengirimkan pesan “1” dengan topik “onindong” dan fungsi OFF mengirimkan pesan “0” dengan topik “offindong”.

2. Fungsi UPLOAD

Untuk fungsi UPLOAD digunakan 5 topik karna terdapat 5 data berbeda yaitu Ruangn, Hari, Mata Kuliah, Jam, dan Kode Dosen seperti pada gambar 4.2.



Gambar 11. *Subscribe* untuk topik pada form *upload*

Pada Gambar 4.8 dilakukan penulisan topik beserta QoS(Quality Of Service) yang digunakan pada MQTTBox, dan setelah tombol subscribe dipencet maka MQTTBox siap menerima pesan dari Android melalui broker.

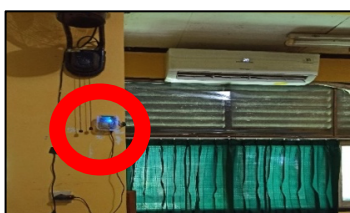


Gambar 12. Hasil pesan yang diterima subscribe form upload

Pada Gambar 4.9 Pesan telah diterima oleh kelima topik dimana untuk data Ruangan mengirimkan pesan "A3-01" dengan topik "jmcdr", untuk data Hari mengirimkan pesan "Senin" dengan topik "jmcdh", untuk data Mata Kuliah mengirimkan pesan "Mata Kuliah" dengan topik "jmcd000", untuk data Jam mengirimkan pesan "10:10" dengan topik "jmcd001" dan untuk data Kode Dosen mengirimkan pesan "OKE" dengan topik "jmcd002".

**4.7. Hasil Pengujian Fungsi Keseluruhan Sistem**

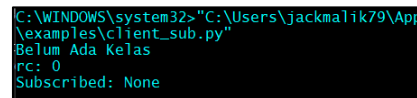
Pengujian fungsi keseluruhan sistem dilakukan di ruang kelas A3-02 PSTI UNRAM. Ada dua skenario pengujian, skenario pertama dilakukan dengan menjalankan sistem untuk mengirim MQTT pada jadwal yang sudah ditentukan dan melakukan pencatatan ke dalam database setelah pesan MQTT berhasil dikirim. Skenario kedua adalah melakukan uji fungsi satu hari penuh sesuai jadwal apakah sistem berjalan semestinya atau tidak dengan memanfaatkan pencatatan seperti pada skenario pertama. Pengujian dilakukan selama beberapa hari yaitu pada hari jumat tanggal 18 Desember 2020, selasa tanggal 29 Desember 2020 dan rabu 30 Desember 2020 di ruang kelas A3-02 PSTI UNRAM dari jam 7:30 sampai jam 16:50. Perangkat keras sistem dipasang pada tempat yang dekat dengan AC dan sumber energi. Yaitu pada lingkaran merah yang ada pada gambar 4.10.



Gambar 13. Letak Penempatan Perangkat Saat Pengujian

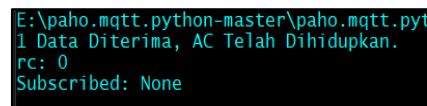
Pada saat sebelum perkuliahan dimulai, program Python pada Task Scheduler (TS) akan berjalan otomatis saat komputer sever sudah mulai menyala. TS akan menjalankan program setiap 10 menit sekali, jika waktu saat TS berjalan bertepatan dengan jadwal perkuliahan maka program akan mengirimkan pesan untuk menghidupkan AC.

Pada program Python yang dijalankan TS ada dua kondisi AC dimatikan yaitu saat jam istirahat dan jam pulang dimana jam istirahat yang dimaksud adalah pada jam 11:40 dan jam pulang adalah 16:50. TS akan terus berjalan 10 menit sekali selama 37 detik selama komputer server menyala, namun tidak selalu mengirim pesan hidup/mati, pesan MQTT hanya dikirimkan sekali saat jam masuk. TS dijalankan 10 menit sekali bertujuan untuk mencegah gagalnya pengiriman pesan MQTT jika ada hal yang diluar kendali terjadi. Pada program Python yang TS jalankan sudah ada kondisi yang mengatur untuk hal tersebut, jika pada saat TS berjalan tidak ada jadwal yang dimulai pada jam tersebut maka akan menampilkan pesan seperti pada gambar 12 dan tidak mengirim pesan MQTT apapun.

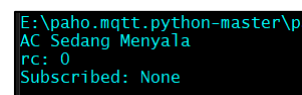


Gambar 14. Task Scheduler saat tidak ada jadwal.

Namun jika ada jadwal yang didapatkan di dalam database maka akan menampilkan pesan seperti pada gambar 13, apabila program sudah berhasil dijalankan pada jam yang sama sebelumnya maka akan mengirim pesan untuk menghidupkan/mematikan AC, lalu menampilkan pesan seperti pada gambar 16.



Gambar 15. Task Scheduler saat ada jadwal pada waktu program dijalankan



Gambar 16. Task Scheduler saat AC sudah pernah dijalankan dijam yang sama.

Salah satu hasil pengujian yang didapatkan dari pengujian TS yaitu pada hari jumat 18 desember 2020 dapat dilihat pada gambar 17

Log Waktu	Log Status
2020-12-18 07:30:01	Hidup
2020-12-18 11:40:03	Mati
2020-12-18 13:50:00	Hidup
2020-12-18 15:10:02	Hidup
2020-12-18 16:50:02	Mati

Gambar 17. Hasil uji Task Scheduler pada jadwal

Pada gambar 4.14 Dilakukan uji coba terhadap jadwal matakuliah pada hari jumat 18 desember 2020. Adapun jadwal perkuliahan yang berlangsung pada hari jumat adalah seperti pada gambar 18

jumat	TBA B	7:30-10:00	BI
	RTI A	13:30-15:10	ARK
	RTI B	15:10--16:50	ARK

Gambar 18. Jadwal matakuliah pada hari jumat

Pengujian dilakukan bersama staf PSTI dan juga beberapa mahasiswa untuk melakukan uji coba fungsi penjadwalan AC dan sensor PIR . Sensor PIR berfungsi untuk mematikan AC jika setelah 30 menit tidak ada pergerakan di dalam ruangan.

Pada minggu berikutnya dilakukan pengujian lagi di ruang A2-03 pada hari selasa 29 Desember 2020 dan rabu 30 desember 2020. Adapun hasil pencatatan log pada hari tersebut dapat dilihat pada gambar 19.

2020-12-29 07:30:01	Hidup	2020-12-30 07:30:01	Hidup
2020-12-29 10:20:02	Hidup	2020-12-30 10:10:02	Hidup
2020-12-29 12:40:02	Mati	2020-12-30 12:40:04	Mati
2020-12-29 13:30:02	Hidup	2020-12-30 13:30:02	Hidup
2020-12-29 15:10:02	Hidup	2020-12-30 16:50:02	Mati
2020-12-29 16:51:23	Mati		

Gambar 19. Hasil log Uji Task Scheduler pada jadwal

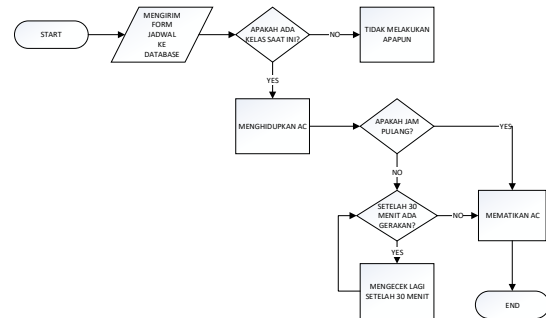
Pada Gambar 4.16 menggunakan jadwal yang berlangsung pada hari selasa dan rabu yang terdaftar pada jadwal yang berlaku. Adapun jadwal yang digunakan dapat dilihat pada gambar 20.

selasa	DATA MINING	7:30-9:10	WWW
	AI B	10:10-12:40	RAM
	KAP B	13:30-15:10	AYH
	PPL B	15:10-16:50	RA
rabu	BIG DATA B	7:30-10:00	WWW
	EP B	10:10-11:50	GSN
	JST B	13:30-15:10	HWJ

Gambar 20. Jadwal matakuliah pada hari selasa dan rabu

Pengujian pada minggu ini dilakukan bersama beberapa mahasiswa untuk melakukan uji coba fungsi penjadwalan AC dan sensor PIR . Sensor PIR berfungsi

untuk mematikan AC jika setelah 30 menit tidak ada pergerakan di dalam ruangan.



Gambar 21. Flowchart Alur Komunikasi Mesin

Berdasarkan hasil yang didapatkan membuktikan program telah beroperasi sesuai yang diharapkan, baik program Python yang dijalankan TS, aplikasi Android dan juga sensor PIR.

## KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil dari penelitian dan juga pengujian yang telah dilaukan, didapatkan hasil kesimpulan sebagai berikut :

1. Berdasarkan hasil pengujian yang telah dilakukan pada Ruang A3-02, Telah berjalan dengan baik untuk fungsi aplikasi Android. Aplikasi sudah berfungsi sebagaimana yang diharapkan, yaitu bisa mengontrol AC kapanpun dan dimanapun , juga dapat melakukan *upload* jadwal matakuliah yang kemudian menjadi data acuan untuk menjalankan AC sesuai jadwal yang sudah ditentukan oleh Program Studi Teknik Informatika UNRAM. Untuk AC yang diuji hanya 1 AC namun jika berkenan menambahkan AC maka perlu tambahan mikro kontroler untuk AC yang lain atau ruangan yang lain. Berbeda halnya jika AC yang dipasang berdekatan maka bisa ditambahkan modul IR *Transmitter* pada mikrokontroler.
2. Berdasarkan hasil pengujian yang telah pada raungan kelas A3-02 maka dapat disimpulkan bahwa sistem telah berjalan dengan baik untuk fungsi penjadwalan AC dan juga sensor PIR. Sistem mampu melakukan otomatisasi sesuai jadwal matakuliah pada gambar 4.16 untuk menghidupkan atau mematikan AC dengan perangkat yang telah dibangun. Sensor PIR yang mendukung untuk mengoptimalkan kinerja dari sistem yang dibangun dengan melakukan deteksi gerak untuk memastikan tidak ada orang di dalam ruangan dan AC dapat dimatikan secara otomatis.

## 5.2 Saran

Untuk menyempurnakan sistem ini untuk kedepannya, ada beberapa saran yang dapat dipertimbangkan untuk menjadi pelengkap dalam pengembangan sistem ini yaitu:

1. Diharapkan sistem yang dibangun memiliki monitoring suhu sekaligus pengendali suhu otomatis dan dapat melihat status AC agar sistem memiliki lebih banyak fitur yang bisa digunakan serta ditambahkan untuk pengamanan penggunaan aplikasi agar tidak sembarang orang bisa mengakses aplikasinya.
2. Diharapkan sistem yang nantinya dibuat dapat memiliki fitur penambahan ruangan agar sistem dapat digunakan di ruangan lain atau ruangan baru yang belum ada pada sistem aplikasi tanpa perlu memprogram ulang sistemnya.

### UCAPAN TERIMA KASIH

Terima kasih saya ucapkan kepada seluruh rekan-rekan yang sudah membantu dalam pembangunan sistem penjadwalan AC ini sehingga dapat terselesaikan sebagaimana mestinya.

### DAFTAR PUSTAKA

- [1] S. Rahayu, Z. Hakim, and N. Septiana, "Sistem Informasi Administrasi Penjualan dan Jasa Air Conditioner ( AC )," *J. Sisfotek Glob.*, vol. 9, no. 2, pp. 79–84, 2019.
- [2] D. Despa, M. A. Muhammad, A. Suriananto, A. Hamni, G. F. Nama, and Y. Martini, "Monitoring dan Manajemen Energi Listrik Gedung Laboratorium Berbasis Internet of Things ( IoT )," *Semin. Nas. Tek. Elektro 2018*, pp. 2–6, 2018.
- [3] B. M. Susanto, E. S. J. Atmadji, and W. L. Brenkman, "Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web," *J. Inform. Polinema*, vol. 4, no. 3, p. 201, 2018, doi: 10.33795/jip.v4i3.207.
- [4] T. Budioko, "Sistem Monitoring Suhu Jarak Jauh Berbasis Internet Of Things Menggunakan Protokol MQTT," *Semin. Ris. Teknol. Inf. tahun*, pp. 353–358, 2016, [Online]. Available: [https://sriti.akakom.ac.id/prosiding/SISTEM Sistem Monitoring Suhu Jarak Jauh Berbasis Internet of Things Menggunakan Protokol MQTT.pdf](https://sriti.akakom.ac.id/prosiding/SISTEM_Sistem_Monitoring_Suhu_Jarak_Jauh_Berbasis_Internet_of_Things_Menggunakan_Protokol_MQTT.pdf).
- [5] "Mobile Operating System Market Share Indonesia", "Mobile Operating System Market Share Worldwide | StatCounter Global Stats," *Www.Gs.Statcounter.Com*, 2020. <https://gs.statcounter.com/os-market-share/mobile/indonesia>.
- [6] M. Pahrurrozi, "Adaptive Classroom Berbasis IoT ( Internet of Things ), Manajemen Penggunaan Air Conditioner ( AC ) Secara Otomatis," *Adapt. Classrom Berbas. IOT (Internet Things), Manaj. Pengguna. Air Cond. Secara Otomatis*, 2020.
- [7] G. Otomo and Wildian, "Perancangan Dimer Lampu Secara Otomatis Berbasis Mikrokontroler Pada Penerangan Dalam Ruangan," *Peranc. Dimer Lampu Secara Otomatis Berbas. Mikrokontroler Pada Penerangan Dalam Ruangan*, vol. 15, no. 4, pp. 186–190, 2014, doi: 10.12777/transmisi.15.4.186-190.
- [8] M. Natsir, D. B. Rendra, and A. D. Y. Anggara, "Implementasi IOT Untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya," *J. PROSISKO Vol. 6 No. 1*, vol. 6, no. 1, 2019.
- [9] A. P. Utomo and N. A. Wirawan, "Perancangan Alat Monitoring Air Conditioner Menggunakan Mikrokontroler Wemos," pp. 44–53, 2018.
- [10] Y. Maulana, "Implementasi Kendali Otomatis Lampu Dan Ac ( Air Conditioning ) Universitas Pasundan Bandung," 2018.
- [11] Tanto and Darmuji, "Penerapan Internet of Things (IoT) Pada Alat Monitoring Energi Listrik," *J. Elektron. List. dan Teknol. Inf. Terap.*, vol. 1, no. 1, pp. 45–51, 2019, [Online]. Available: <https://ojs.politeknikjambi.ac.id/elti>.
- [12] H. B. Santoso, S. Prajogo, and S. P. Mursid, "Pengembangan Sistem Pemantauan Konsumsi Energi Rumah Tangga Berbasis Internet of Things (IoT)," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 6, no. 3, p. 357, 2018, doi: 10.26760/elkomika.v6i3.357.
- [13] Y. Efendi, "Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile," *J. Ilm. Ilmu Komput.*, vol. 4, no. 2, pp. 21–27, 2018, doi: 10.35329/jiik.v4i2.41.
- [14] S. R. Nurhalimah, S. Suhartono, and U. Cahyana, "Jurnal Riset Pendidikan Kimia ARTICLE," *J. Ris. Pendidik. Kim.*, vol. 7, no. 1, pp. 38–51, 2017, doi: <https://doi.org/10.21009/JRPK.072.10>.
- [15] A. F. Silvia, E. Haritman, and Y. Muladi, "Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino Dan Android," *Electrans*, vol. 13, no. 1, pp. 1–10, 2014.
- [16] S. J. Sokop, D. J. Mamahit, and S. R. U. A. Sompie, "Trainer Periferan Antarmuka Berbasis

- Mikrokontroler Arduino Uno," *E-Journal Tek. Elektro Dan Komput.*, vol. 5, no. 3, pp. 13–23, 2016, doi: 10.35793/jtek.5.3.2016.11999.
- [17] F. Supegina and E. J. Setiawan, "Rancang Bangun Iot Temperature Controller Untuk Enclosure Bts Berbasis Microcontroller Wemos Dan Android," *J. Teknol. Elektro, Univ. Mercu Buana*, vol. 8, no. 2, pp. 145–150, 2017.
- [18] R. H. Zain, "Sistem Keamanan Ruang Menggunakan Sensor Passive Infra Red (PIR) Dilengkapi Kontrol Penerangan Pada Ruang Berbasis Mikrokontroler Atmega8535 Dan Real Time Clock Ds1307," *J. Teknol. Inf. dan Pendid.*, vol. 6, no. 1, pp. 45–54, 2013.
- [19] Yusniati, "Penggunaan Sensor Infrared Switching Pada Motor DC Satu Fasa," *J. Electr. Technol.*, vol. Vol. 3, No, pp. 90–96, 2018.
- [20] P. I. A. Guna, I. M. A. Suyadnya, and I. G. A. P. R. Agung, "Sistem Monitoring Penetasan Telur Penyu Menggunakan Mikrokontroler NodeMCU ESP8266 dan Protokol MQTT dengan Notifikasi Berbasis Telegram Messenger," *J. Comput. Sci. Informatics Eng.*, vol. 2, no. 2, p. 80, 2018, doi: 10.29303/jcosine.v2i2.135.
- [21] H. A. Rochman, R. Primananda, and H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 6, pp. 445–455, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id>.