

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

Tugas Akhir
untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



OLEH :

MUHAMMAD MALIK SAPUTRA

F1D016059

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2021

TUGAS AKHIR

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

Oleh:

MUHAMMAD MALIK SAPUTRA

F1D 016 059

Telah diperiksa oleh Tim Pembimbing:

1. Pembimbing Utama



Tanggal: 24/01/2021

Dr. Eng. I Gede Putu Wirarama W.W., S.T, M.T.
NIP. 19840919 201803 1 001

2. Pembimbing Pendamping



Tanggal: 24/01/2021

Ariyan Zubaidi, S.Kom., M.T.
NIP. 19860913 201504 1 001

Mengetahui
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Prof. I GP Suta Wijaya, S.T.,M.T.,D.Eng.
NIP. 19731130 200003 1 001

TUGAS AKHIR

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

Oleh:

MUHAMMAD MALIK SAPUTRA

F1D 016 059

Telah diujikan di depan penguji

Pada tanggal 12 Januari 2021

Dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1
Program Studi Teknik Informatika

Susunan Tim Penguji:

1. Penguji 1



Tanggal: 25/01/2021

Andy Hidayat Jatmika, S.T., M.Kom.

NIP. 198312092012121001

2. Penguji 2



Tanggal: 24/01/2021

Ahmad Zafrullah M., S.T., M.Eng.

NIP.

3. Penguji 3



Tanggal: 25/01/2021

Arik Aranta, S.Kom., M.Kom.

NIP. 199402202019031004

Mataram, 26 Januari 2021

Dekan Fakultas Teknik

Universitas Mataram



Akmaluddin, ST., M.Sc Eng., Ph.D.

NIP. 196812311994121001

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan bahwa tugas akhir ini dengan judul “Sistem Penjadwalan *Air Conditioner* (AC) Ruangan Berdasarkan Jadwal Matakuliah Menggunakan ESP8266, PIR Sensor dan Android” sepenuhnya adalah karya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan Saya tidak melakukan penjiplakan atau pengutipan dengan cara yang tidak sesuai dengan etika keilmuan yang berlaku. Atas pernyataan ini, Saya siap menanggung resiko/sanksi yang dijatuhkan kepada Saya pribadi apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya ini, atau ada klaim dari pihak lain terhadap keaslian karya ini.

Mataram, 26 Januari 2021

Yang membuat pernyataan,

Muhammad Malik Saputra

PRAKATA

Assalamualaikum Wr. Wb.

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-NYA sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Sistem Penjadwalan *Air Conditioner* (AC) Ruang Berdasarkan Jadwal Matakuliah Menggunakan ESP8266, PIR Sensor dan Android” tepat pada waktunya. Pada kesempatan ini pula, penulis menghaturkan terimakasih yang sebesar-besarnya kepada seluruh pihak yang telah mendukung agar terselesaikannya Tugas Akhir ini.

Penulis tentu menyadari Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan di dalamnya. Untuk itu, diharapkan kritik serta saran dari pembaca untuk Tugas Akhir ini, agar menjadi lebih baik lagi. Penulis juga mengucapkan banyak-banyak terimakasih kepada semua pihak yang telah membantu penulis sehingga Tugas Akhir ini dapat selesai.

Demikian yang bisa disampaikan, sekali lagi terimakasih atas semua pihak yang telah membantu di dalam penyelesaian Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

Wassalamualaikum Wr. Wb.

Mataram, 21 Juli 2020

Penulis

UCAPAN TERIMA KASIH

Penulis menyadari bahwa terselesaikannya Tugas Akhir ini tentunya bukan hanya dari usaha penulis saja. Tugas Akhir ini bisa selesai tepat waktu tentunya berkat dukungan dari semua pihak yang terlibat juga. Oleh karena itu, pada kesempatan ini penulis menghaturkan terimakasih yang sebesar-besarnya kepada:

1. Orang Tua dan keluarga yang selalu memberikan dukungan dari segala aspek kehidupan selama ini.
2. Bapak Dr.Eng. I Gde Putu Wirarama WW., S.T, M.T. selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
3. Bapak Ariyan Zubaidi, S.Kom.,M.T. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis dengan amat teliti selama penyusunan Tugas Akhir sehingga dapat selesai dengan baik.
4. Neyla Vista Maramy, Imam Haraq, yang tidak pernah lelah menjadi pendukung dan teman diskusi selama pengerjaan Tugas Akhir.
5. Staf dan Kampus Program Studi Teknik Informatika Universitas Mataram selaku tempat dilakukannya penelitian dan uji coba sistem.
6. Responden Ketika pengumpulan kebutuhan dan penelitian yang telah meluangkan waktunya sehingga saran dan masukannya berguna untuk pengembangan sistem. Juga “penyu” yang selalu menjadi penyusah sekaligus *refresher* saat pengerjaan Tugas Akhir .
7. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan do’a dan dukungan baik moril maupun materi sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik

Semoga Tuhan Yang Maha Kuasa selalu memberikan rahmat dan hidayah-Nya dan memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis.

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PRAKATA.....	v
UCAPAN TERIMA KASIH.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
DAFTAR KODE SUMBER.....	xi
ABSTRAK.....	xii
ABSTRACT.....	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat.....	4
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1. Tinjauan Pustaka.....	6
2.2. Dasar Teori.....	9
2.2.1. <i>Air Conditioner (AC)</i>	9
2.2.2. <i>Internet of Things (IoT)</i>	9
2.2.3. <i>Android</i>	10
2.2.4. <i>Mikrokontroler</i>	10
2.2.5. <i>Wemos</i>	11
2.2.6. <i>Sensor Passive Infrared (PIR)</i>	12
2.2.7. <i>IR Transmitter</i>	13
2.2.8. <i>IR Receiver</i>	13
2.2.9. <i>Sistem Monitoring</i>	14
2.2.10. <i>Protokol Message Queue Telemetry Transport (MQTT)</i>	14
BAB III METODE PENELITIAN.....	15
3.1. Rencana Pelaksanaan.....	15
3.2. Analisis Kebutuhan.....	16
3.3. Rancangan Arsitektur Sistem.....	17
3.4. Rancangan Perangkat Keras.....	18
3.5. Rancang Perangkat Lunak.....	19

3.5.1.	Desain Sistem	19
3.5.2.	Rancangan Komunikasi MQTT	21
3.6.	Implementasi Sistem	22
3.7.	Pengujian dan Evaluasi Sistem	22
3.8.	Dokumentasi dan Laporan.....	23
BAB IV HASIL DAN PEMBAHASAN.....		24
4.1	Realisasi Sistem	24
4.1.1	Realisasi Penyusunan Perangkat Keras	24
4.1.2	Realisasi Pembangunan Aplikasi Android.....	25
4.1.3	Realisasi Pembangunan Komunikasi Data	27
4.1.4	Realisasi Pembangunan Program Pada Mikrokontroler	32
4.2	Pengujian dan Evaluasi Sistem	36
4.2.1	Hasil Pengujian Sistem Aplikasi Android	37
4.2.2	Hasil Pengujian Fungsi Keseluruhan Sistem.....	39
BAB V		43
5.1	Kesimpulan	43
5.2	Saran	43
DAFTAR PUSTAKA		44

DAFTAR GAMBAR

Gambar 2.1 Konsep IoT[13]	10
Gambar 2.2 Mikrokontroler Arduino Mega 2560[8]	11
Gambar 2.3 Bentuk Fisik Dari Wemos D1[17].....	11
Gambar 2.4 Bentuk Fisik PIR Sensor[6].....	12
Gambar 2.5 Diagram Internal Rangkaian Sensor PIR	13
Gambar 2.6 Arah Jangkauan Sensor PIR.....	13
Gambar 3.1 <i>Flowchart</i> Tahapan Penelitian.....	15
Gambar 3.2 Rancangan Arsitektur Sistem	17
Gambar 3.3 Rancangan Perangkat Keras	19
Gambar 3.5 Rancangan <i>Entity Relationship Diagram</i>	20
Gambar 3.6 Rancangan Beranda dan Halaman Upload Jadwal Aplikasi Android.....	21
Gambar 3.7 Rancangan Komunikasi MQTT	22
Gambar 4.1 Realisasi Perangkat Keras	24
Gambar 4.2 Tampilan Menu Awal Aplikasi	25
Gambar 4.3 Tampilan Menu Jadwal	26
Gambar 4.4 Tampilan Menu <i>Form</i> Tambah dan Edit Jadwal Aplikasi	26
Gambar 4.5 Tampilan Awal MQTT Box	37
Gambar 4.6 Contoh MQTTBox Sebagai <i>Subscriber</i>	38
Gambar 4.7 <i>Subscribe</i> Untuk Tombol ON dan OFF.....	38
Gambar 4.8 Hasil Pesan Yang Diterima <i>Subscribe</i> ON dan OFF.....	38
Gambar 4.9 <i>Subscribe</i> Untuk Topik Pada Form <i>Upload</i>	39
Gambar 4. 10 Hasil Pesan Yang Diterima <i>Subscribe</i> Form <i>Upload</i>	39
Gambar 4.12 Letak Penempatan Perangkat Saat Pengujian	40
Gambar 4.12 <i>Task Scheduler</i> Saat Tidak Ada Jadwal	41
Gambar 4.13 <i>Task Scheduler</i> Saat Ada Jadwal Pada Waktu Program Dijalankan	41
Gambar 4.15 Hasil Uji <i>Task Scheduler</i> Pada Jadwal	41
Gambar 4.16 Jadwal Matakuliah Pada Hari Jumat	41
Gambar 4.18 Jadwal Matakuliah Pada Hari Selasa dan Rabu	42
Gambar 4.119 <i>Flowchart</i> Alur Komunikasi Mesin	42

DAFTAR TABEL

Tabel 3.1 <i>Database</i> Matakuliah	20
Tabel 3.2 <i>Database</i> Dosen	20
Tabel 3.3 <i>Database</i> Jadwal	20
Tabel 3.4 <i>Database</i> Semester.....	20
Tabel 3.5 <i>Database</i> Mengajar	21
Tabel 4.1 Hasil Pengujian Aplikasi Android	37

DAFTAR KODE SUMBER

Kode Sumber 4.1 <i>Script</i> modul/ <i>package</i>	27
Kode Sumber 4.2 <i>Script</i> deklarasi dari <i>database</i>	27
Kode Sumber 4.3 <i>Script</i> deklarasi fungsi koneksi dan <i>subscribe</i>	28
Kode Sumber 4.4 <i>Script</i> deklarasi fungsi <i>input</i> data ke dalam <i>database</i>	28
Kode Sumber 4.5 <i>Script</i> deklarasi fungsi mengambil pesan/ <i>message</i> melalui MQTT	29
Kode Sumber 4.6 <i>Script</i> deklarasi fungsi untuk mengambil data dari <i>database</i>	30
Kode Sumber 4.7 Deklarasi variabel lanjutan.....	30
Kode Sumber 4.8 <i>Script</i> kondisi untuk mengirimkan pesan MQTT sesuai jadwal	31
Kode Sumber 4.9 Kondisi untuk mengontrol AC dan meng- <i>input</i> laporan	32
Kode Sumber 4.10 Deklarasi variabel mikrokontroler	33
Kode Sumber 4.11 Fungsi Wifi pada program mikrokontroler	34
Kode Sumber 4.12 Fungsi untuk mengkoneksikan Kembali pada mikrokontroler	34
Kode Sumber 4.13 Fungsi <i>subscribe</i> pada program di mikrokontroler	35
Kode Sumber 4.14 Fungsi <i>setup</i> program mikrokontroler.....	35
Kode Sumber 4.15 Fungsi <i>loop</i> pada program di mikrokontroler	36

ABSTRAK

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 seringkali penggunaannya tidak terorganisir seperti membiarkan AC tetap menyala bahkan setelah ruang kelas kosong atau tidak ada kegiatan yang menyebabkan tidak efisien dalam menggunakan AC yang juga dapat menimbulkan kerusakan pada AC tersebut. Oleh karena itu perlu adanya sistem yang mengontrol penjadwalan AC secara otomatis yang lebih efisien. Hal ini dapat diwujudkan dengan teknologi *Internet of Things*(IoT). Maka perlu dibangun sistem yang akan membuat AC otomatis menyala sesuai jadwal kuliah menggunakan mikrokontroler(ESP8266 Wemos D1) dengan *infrared transmitter* untuk mengendalikan AC dan deteksi gerak yang akan menggunakan protokol *Message Queueing Telemetry Transport* (MQTT) sebagai broker. Sensor gerak *Passive Infrared*(PIR) *Sensor* akan mendeteksi gerakan pada ruang kelas yang menjadi kondisi untuk mematikan AC jika setelah 30 menit tidak ada pergerakan didalam ruangan. Untuk pengoperasiannya menggunakan *smartphone* android sekaligus sebagai antara muka untuk memasukkan jadwal perkelas hanya staf PSTI UNRAM yang akan mengoperasikan aplikasinya. Dengan begitu AC bisa otomatis menyala sesuai jadwal mata kuliah dan bisa mati secara otomatis setelah 30 menit tidak ada orang di dalam ruangan.

Keywords : IOT, PIR, ANDROID, ESP8266, WEMOS, MQTT, AC

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 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, ANDROID, ESP8266, WEMOS, MQTT, AC

BAB I

PENDAHULUAN

1.1. Latar Belakang

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 terorganisir seperti membiarkan AC tetap menyala bahkan setelah ruang kelas kosong atau tidak ada kegiatan yang menyebabkan tidak efisien dalam menggunakan AC yang juga dapat menimbulkan kerusakan pada AC tersebut.

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 survei 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.

Oleh karena itu perlu adanya sistem yang mengontrol penjadwalan AC secara otomatis yang lebih efisien agar tidak terjadi hal-hal seperti AC tetap menyala saat tidak ada orang bahkan sampai esok harinya. Hal ini dapat diwujudkan dengan teknologi *Internet of Things*(IoT). Untuk kasus mahasiswa yang iseng menyalakan AC akan diatasi pada penelitian yang lain terkait kunci pintu otomatis. Pada penelitian ini akan berfokus pada otomasisasi penggunaan AC berdasarkan jadwal matakuliah.

Teknologi saat ini dapat dibidang mengalami perkembangan yang cukup pesat ditandai dengan banyaknya teknologi yang dapat dioperasikan melalui jarak jauh menggunakan bantuan alat-alat tertentu. Salah satu contoh perkembangan teknologi saat ini yaitu *Internet of Things*(IoT) yang dapat digunakan dalam berbagai macam alat-alat elektronik untuk mengontrolnya secara *real time* dari sebuah perangkat yang bahkan memiliki jarak yang sangat jauh dengan bantuan internet tentunya. *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].

Penggunaan IoT semakin berkembang dalam waktu singkat, hal ini dikarenakan oleh semakin berkembangnya teknologi informasi. Hal ini menyebabkan keterlibatan banyak *device* yang saling terkoneksi dengan sensor yang terpasang pada lingkungan. Sehingga dengan adanya banyak *device* maka menyebabkan permasalahan interoperabilitas dari masing-masing alat[3]. Pada konsep IoT diperlukan protokol untuk dapat mengoperasikannya dengan baik dan benar, ada banyak sekali protokol komunikasi pada IOT dengan berbagai macam kelebihan dan kekurangan masing-masing, salah satu di antara protokol ini adalah (*Message Queue Telemetry Transport*)MQTT, (*Constrained Application Protocol*)CoAP, (*Extensible Messaging and Presence Protocol*)XMPP, dan berbagai macam protokol yang menggunakan (*HyperText Transfer Protocol*)HTTP. Pada pengembangan ini, protokol yang akan digunakan adalah *Message Queue Telemetry Transport*(MQTT) karena lebih cepat di banding HTTP untuk mengirim data yang kecil juga dapat bekerja dengan energi dan media penyimpanan yang minimum, selain itu juga lebih simpel dalam pengoperasiannya dalam suatu sistem.

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(PIR) Sensor*, untuk mendeteksi gerakan pada ruang kelas yang akan menjadi kondisi untuk mematikan AC jika setelah 30 menit tidak ada pergerakan didalam ruangan. Mikrokontroler yang digunakan adalah wemos D1 yang merupakan turunan dari ESP8266 karena memiliki wifi dan harga yang murah, selain wemos atau ESP8266 ada beberapa mikro kontroler lain yg juga bisa digunakan yaitu Arduino UNO, dan ESP32, namun keduanya memiliki harga yang jauh lebih mahal dibandingkan Wemos D1. Pada penelitian ini digunakan *smartphone(mobile apps)* untuk meningkatkan pengalaman pengguna dalam pengoperasian dikarenakan praktisnya menjalankan aplikasi yg terdedikasi dibandingkan dengan pengoperasian berdasarkan *web browser*, atau *desktop app* yang mengharuskan pengguna menghidupkan PC terlebih dahulu, juga agar lebih aman menggunakan *smartphone(mobile apps)* sebagai antar muka memasukkan jadwal perkelas karena yang dapat mengakses aplikasinya hanya staf PSTI UNRAM. Sistem Operasi dari *smartphone* yang digunakan adalah android karena penelitian ini difokuskan ke *smartphone* android saja dan menurut data penggunaan sistem operasi android di salah satu *website statcounter* dalam 12 bulan terakhir di Indonesia mencapai 92.54% di bandingkan iOS yang hanya 7.25%[5]. Selain itu juga *Smartphone* yang digunakan oleh semua staf PSTI UNRAM adalah Android.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat diambil rumusan masalah sebagai berikut :

Bagaimana melakukan manajemen penggunaan AC ruangan kelas berdasarkan jadwal matakuliah di PSTI UNRAM berbasis IoT?

1.3. Batasan Masalah

Adapun batasan masalah yang dihadapi yaitu :

1. Implementasi dan pengujian sistem dilakukan pada ruang kelas A3-02 PSTI UNRAM.

2. Otomatisasi yang dilakukan yaitu dapat menghidupkan, dan mematikan AC sesuai dengan jadwal kuliah.
3. Tidak dilakukannya monitoring jumlah penggunaan daya.
4. Sistem yang dirancang tidak menangani proses penyesuaian suhu ruangan.
5. Hanya melakukan pengujian pada 1 AC di ruangan A3-02 PSTI UNRAM
6. API minimum yang digunakan untuk aplikasi android adalah level 16 dan API maksimum level 28.

1.4. Tujuan Penelitian

Tujuan pada penelitian ini adalah sebagai berikut :

Membangun sistem penjadwalan AC ruangan kelas berdasarkan jadwal matakuliah di PSTI UNRAM berbasis IoT.

1.5. Manfaat

Adapun manfaat dari penelitian tugas akhir ini adalah:

1. Mengefisienkan penggunaan AC pada ruang kuliah PSTI UNRAM menjadi lebih terorganisir menggunakan konsep IoT.
2. Dapat mengendalikan AC secara *real time* dengan menggunakan android.

1.6. Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam penelitian tugas akhir ini menjadi 5 bab sebagai berikut :

1. Bab I. Pendahuluan

Bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.

2. Bab II. Tinjauan Pustaka dan Dasar Teori

Bab ini memuat tentang tinjauan pustaka yang menjabarkan hasil penelitian yang berkaitan dengan penelitian ini dan dasar teori yang menjabarkan teori-teori penunjang yang berhubungan dengan penelitian ini.

3. Bab III. Metode Perancangan

Memuat tentang metode perancangan, mulai dari pelaksanaan penelitian, diagram alir penelitian, menentukan alat dan bahan, lokasi penelitian, dan langkah-langkah penelitian.

4. Bab IV. Hasil dan Pembahasan

Memuat tentang hasil dan pembahasan yang diperoleh berdasarkan hasil pengukuran dan pelaksanaan.

5. Bab V. Penutup

Memuat tentang kesimpulan dan saran berdasarkan hasil pembahasan yang telah diperoleh.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Berikut merupakan beberapa tinjauan pustaka dari penelitian terkait yang telah dilakukan sebelumnya, sebagai bahan acuan untuk melakukan penelitian ini.

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 rancang-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). Jarak waktu respon dari sensor PIR KC7783R telah dicobakan, dimana sensor hanya dapat mendeteksi objek selama 5,37 detik, namun dapat diatasi dengan menggunakan program yang ditanamkan ke dalam mikrokontroler AT89S51. Sensor membutuhkan waktu pemanasan selama 25,52 detik. *Relay* digunakan 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], terkait ialah Implementasi IoT Untuk Sistem Kendali AC Otomatis Pada Ruang Kelas Di Universitas Serang Raya. 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 dan meningkatnya biaya tagihan listrik yang harus dibayarkan. Hal ini disebabkan karena seringnya AC beroperasi secara terus menerus dan tidak dimatikan apabila ruangan kelas kosong atau tidak ada kegiatan belajar mengajar pada ruangan kelas tersebut. 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. Di kampus Universitas Serang Raya telah tersedia slot kabel LAN RJ45 yang telah menjangkau ke semua ruangan kelas yang ada di area kampus Universitas Serang Raya. Hal ini memunculkan ide untuk memanfaatkan slot RJ45 tersebut untuk dibuat sistem kendali AC dari jarak jauh. Dalam penelitian ini digunakan microcontroller ATmega 2560 sebagai perangkat pengendali, slot RJ45 sebagai media infrastruktur konektivitas jaringan dan smartphone atau komputer sebagai perangkat penyedia user interface untuk memonitor suhu, kelembaban, status AC dan jumlah orang yang berada di ruang kelas[8].

Pada penelitian[8], menggunakan objek yang sama berupa AC yang akan dioperasikan melalui konsep IoT, namun terdapat perbedaan yang mendasar dengan penelitian ini karena menggunakan mikrokontroler ATmega 2560, memanfaatkan slot RJ45 sebagai media infrastruktur dan sensor suhu DS18B20 untuk memantau suhu pada ruangan. 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, kelembapan, 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. Pengukuran parameter temperatur dan kelembapan menggunakan modul AM2302, status *power air*

conditioner dengan menggunakan modul KY-003, keberadaan manusia dengan menggunakan modul HC-SR501, dan mencatat waktu *realtime* dengan menggunakan modul DS3231. Alat monitoring dirancang agar dapat menyimpan secara internal dengan modul SD card dan menampilkan data sensor melalui portal *Thingspeak*. Alat monitoring air conditioner diuji coba dengan berbagai kondisi termal ruangan. 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. Konsep ini juga memungkinkan pembuat *smart home* sistem untuk mengembangkan sistem yang telah dibuat dengan menambah *node* yaitu *relay* dan sensor yang terkoneksi ke *microcontroller*, dengan satu server terpusat yaitu Raspberry PI. OpenHAB adalah *framework smart home system* yang sesuai dengan konsep *Internet of Things*, dengan menggunakan *framework* OpenHAB maka pengguna *smart home system* menjadi lebih mudah karena tidak perlu lagi membuat aplikasi server dan aplikasi pengontrol. 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 ialah sensor suhu DHT11, IR receiverTSOP

38238, IR transmitter serta OLED *display* dan menggunakan Raspberry Pi sebagai servernya. 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.

2.2. Dasar Teori

Dasar teori tentang konsep dan alat yang digunakan dalam perancangan dan pembuatan sistem pada penelitian ini akan dijelaskan pada subbab berikut ini:

2.2.1. Air Conditioner (AC)

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 menyejukkan suhu atau temperatur udara dalam suatu ruangan yang disesuaikan dengan kondisi tubuh penghuni ruangan tersebut[1].

2.2.2. 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 di sekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan seperti internet. Berawal dari *Auto-ID Center*, teknologi yang berbasis pada *Radio Frequency Identification* (RFID) yang merupakan identifikasi kode produk elektronik yang bersifat unik ini kemudian berkembang menjadi teknologi bahwa pada setiap benda dapat memiliki alamat *Internet Protocol* (IP). Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet[11].

Teknologi IoT memungkinkan pengendalian objek dari jarak jauh di seluruh infrastruktur jaringan yang ada dan mampu menciptakan peluang untuk integrasi antara dunia fisik dan sistem digital berbasis *cyber* sehingga dapat meningkatkan efisiensi, akurasi, dan manfaat ekonomi. Setiap objek/ *things* mampu diidentifikasi

melalui sistem komputasi yang tertanam dan mampu beroperasi dalam infrastruktur internet yang ada[12].



Gambar 2.1 Konsep IoT[13]

2.2.3. 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].

Android adalah platform *open source* yang komprehensif dan dirancang untuk *mobile devices*. Dikatakan komprehensif karena Android menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. Sistem Android menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan. 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].

2.2.4. Mikrokontroler

Mikrokontroler (bahasa Inggris: *microcontroller*) merupakan sistem mikroprosesor lengkap yang terkandung di dalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena di dalam sebuah mikrokontroler umumnya telah terdapat komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O, bahkan ada beberapa jenis

mikrokontroler yang memiliki fasilitas ADC, PLL, EEPROM dalam satu kemasan, sedangkan di dalam mikroprosesor umumnya hanya berisi CPU saja[16].



Gambar 2.2 Mikrokontroler Arduino Mega 2560[8]

2.2.5. 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. Wemos menggunakan *chipset* CH340 yang digunakan untuk mengubah USB menjadi serial *interface*. Sebagai contohnya, adalah aplikasi USB *converter to IrDa* atau aplikasi USB *converter to printer*. Dalam mode serial *interface*, CH340 mengirimkan sinyal penghubung yang umum digunakan untuk MODEM. CH340 digunakan untuk memperbesar *asynchronous serial interface* komputer atau mengubah perangkat serial *interface* umum untuk berhubungan dengan bus USB secara langsung. *Driver* atau *software* dari CH340 dapat diunduh secara gratis. Kabel USB seperti kabel data *micro* USB OTG. Wemos menggunakan *software* Arduino baik IDE, *Library*, maupun *command* yang terdapat pada Arduino. Hanya *board* Arduino Uno harus diganti dengan Wemos D1 R1[9].



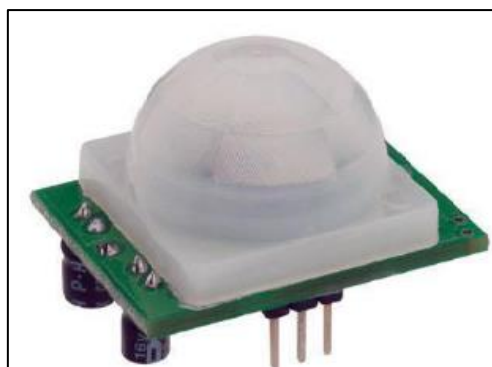
Gambar 2.3 Bentuk Fisik Dari Wemos D1[17].

Bahasa pemrograman sama seperti arduino sehingga fungsi setup pin dan fungsi *loop* sama . Berikut ini adalah spesifikasi teknis dari mikrokontroler Wemos D1 R1[9]:

1. 11 *digital input/output pins*,
2. 1 *analog input* (maksimal *input* 3.2V)
3. *Micro USB connection*
4. *Power jack, 9-24V power input*.
5. Kompatibel dengan Arduino
6. Kompatibel dengan nodemcu
7. *Operating Voltage: 3.3V*
8. *Digital I/O Pins: 11*
9. *Analog Input Pins: 1(Max input: 3.2V)*
10. *Clock Speed: 80MHz/160MHz*
11. *Flash: 4M bytes*
12. Panjang: 68.6mm
13. Lebar: 53.4mm
14. Bobot: 25g

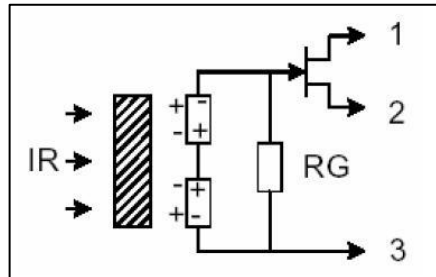
2.2.6.Sensor Passive Infrared (PIR)

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].

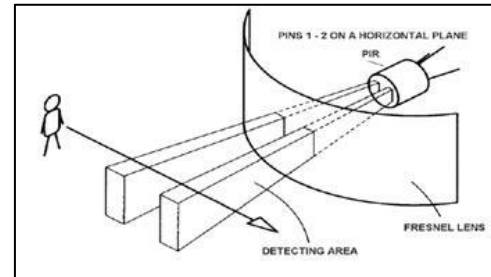


Gambar 2.4 Bentuk Fisik PIR Sensor[6].

Sensor PIR mempunyai dua *element 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].



Gambar 2.5 Diagram Internal Rangkaian Sensor PIR



Gambar 2.6 Arah Jangkauan Gelombang Sensor PIR.

2.2.7. IR Transmitter

Infrared transmitter merupakan suatu model pengirim data melalui gelombang infra merah dengan frekuensi *carrier* sebesar 38 kHz. Modul ini dapat difungsikan sebagai *output* dalam aplikasi transmisi data nirkabel seperti *robotic*, sistem pengamanan, dan sebagainya. Pemancar yang digunakan pada sistem ini terdiri atas sebuah *Light Emitting Diode (LED)*. LED adalah suatu bahan semi konduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju. LED infra merah jenis diode yang memancarkan cahaya infra merah, aplikasi sederhana penggunaan LED infra merah ini adalah pada remot TV. LED infra merah pada dasarnya adalah diode PN *silicon* biasa yang dikemas dalam kotak transparan[19].

2.2.8. IR Receiver

Infrared receiver merupakan suatu modul penerima data melalui gelombang infra merah dengan frekuensi *carrier* sebesar 38kHz. Modul ini dapat difungsikan sebagai input dalam aplikasi transmisi data nirkabel seperti robotik, sistem pengaman, dan sebagainya. *Receiver* (penerima) yang digunakan untuk sensor infra merah adalah jenis *photo otransistor*, yaitu jenis transistor bipolar yang menggunakan kontak (*junction*) *base-collector* untuk menerima atau mendekteksi cahaya dengan gain internal yang dapat menghasilkan sinyal analog maupun digital. *Photo transistor* merupakan salah satu komponen yang berfungsi sebagai *detector* cahaya yang dapat

mengubah efek cahaya menjadi sinyal listrik. Karena itu *photo transistor* termasuk dalam *detector optic*[19].

2.2.9. Sistem *Monitoring*

Sistem *monitoring* adalah sistem yang dapat melakukan pemantauan, pengamatan, dan pengawasan. *Monitoring* bertujuan untuk memberikan suatu informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. *Monitoring* dapat didefinisikan siklus kegiatan yang mencakup pengumpulan informasi, peninjauan ulang atau *review*, *report* dan tindakan atas informasi suatu proses yang sedang diimplementasikan[20].

2.2.10. Protokol *Message Queue Telemetry Transport (MQTT)*

Message Queue Telemetry Transport (MQTT) adalah sebuah protokol komunikasi data *machine to machine* (M2M) yang berada pada layer aplikasi, MQTT bersifat *lightweight message* artinya MQTT berkomunikasi dengan mengirimkan data pesan yang memiliki *header* berukuran kecil yaitu hanya sebesar 2 *bytes* untuk setiap jenis data, sehingga dapat bekerja di dalam lingkungan yang terbatas sumber dayanya seperti kecilnya *bandwidth* dan terbatasnya sumber daya listrik, selain itu protokol ini juga menjamin terkirimnya semua pesan walaupun koneksi terputus sementara, protokol MQTT menggunakan metode *publish/subscribe* untuk metode komunikasinya.

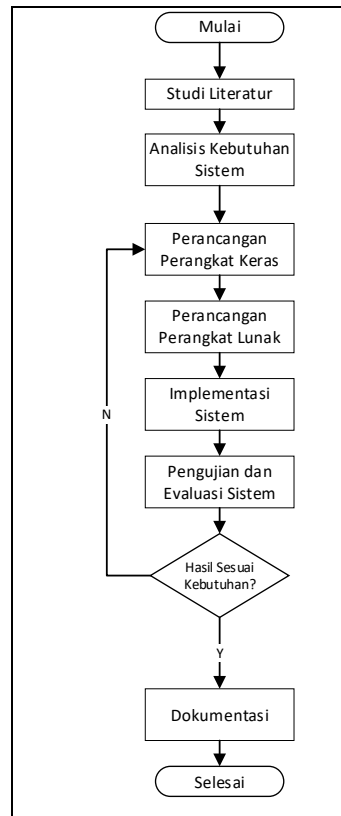
Publish/subscribe sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana pengirim data disebut *publisher* dan penerima data disebut dengan *subscriber*, metode *publish/subscribe* memiliki beberapa kelebihan salah satunya yaitu *loose coupling* atau *decouple* dimana berarti antara *publisher* dan *subscriber* tidak saling mengetahui keberadaannya, terdapat 3 buah *decoupling* yaitu *time decoupling*, *space decoupling* dan *synchronization decoupling*, *time decoupling* adalah sebuah kondisi dimana *publisher* dan *subscriber* tidak harus saling aktif pada waktu yang sama, *space decoupling* adalah dimana *publisher* dan *subscriber* aktif di waktu yang sama akan tetapi antara *publisher* dan *subscriber* tidak saling mengetahui keberadaan dan identitas satu sama lain, dan yang terakhir adalah *synchronization decoupling* kondisi dimana pengaturan *event* baik itu penerimaan atau pengiriman pesan di sebuah *node* hingga tidak saling mengganggu satu sama lain[21].

BAB III

METODE PENELITIAN

3.1. Rencana Pelaksanaan

Adapun rencana pelaksanaan penelitian sistem penjadwalan AC ruangan berdasarkan matakuliah secara otomatis dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* Tahapan Penelitian

Pada Gambar 3.1 memperlihatkan alur dari rencana pelaksanaan penelitian sistem penjadwalan AC menggunakan jadwal matakuliah. Penjelasan untuk masing-masing proses pada Gambar 3.1 adalah sebagai berikut:

1. Pada tahap studi literatur, dilakukan pengumpulan studi kasus yang berkaitan dengan topik penelitian yang dilakukan seperti penelitian terkait AC dengan IoT, Mikrokontroler, PIR sensor, Android dan penelitian-penelitian yang sudah dilaksanakan sebelumnya yang masih terkait dengan penelitian yang akan dilaksanakan.

2. Pada tahap analisis kebutuhan sistem, dilakukan analisa kebutuhan dari sistem yang dibangun, yaitu menjelaskan apa saja perangkat yang dibutuhkan dalam proses perancangan dan pembangunan sistem yang sedang dilaksanakan.
3. Pada tahap perancangan perangkat keras, dilakukan perancangan terhadap keseluruhan rangkaian perangkat keras sistem berupa arsitektur dan rangkaian elektronika yang akan dibangun.
4. Pada tahap perancangan perangkat lunak, akan berkaitan dengan perancangan protokol MQTT dan sistem berbasis android sebagai media untuk mengontrol AC dari perangkat Andorid.
5. Pada tahap implementasi, dilakukan penyusunan perangkat dan penerapan protokol MQTT dan penyusunan aplikasi-aplikasi yang dibutuhkan oleh sistem untuk diimplementasikan nantinya.
6. Pada tahap pengujian dan evaluasi sistem, dilakukan pengujian terhadap sistem yang telah dibangun dan melakukan evaluasi dari hasil pengujian yang sudah dilakukan. Jika sistem berjalan sesuai dengan kebutuhan maka akan dilanjutkan ke tahap berikutnya, namun jika sistem belum berjalan sesuai dengan kebutuhan maka akan dilakukan perbaikan dari perancangan perangkat keras kembali.
7. Pada tahap dokumentasi, akan dibuat laporan dari hasil pengujian dan evaluasi sistem yang telah dilakukan.

3.2. Analisis Kebutuhan

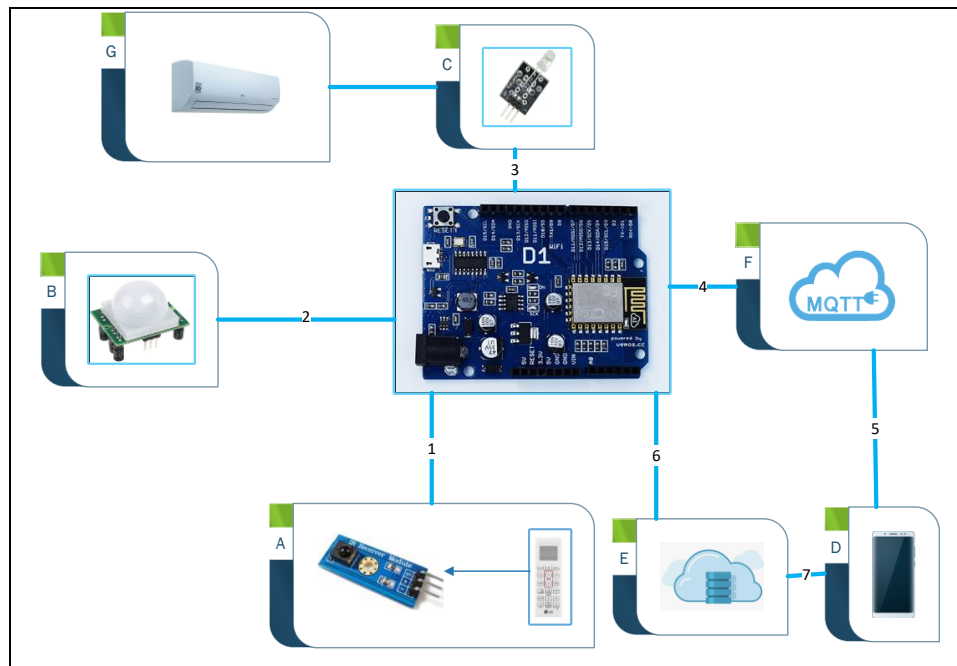
Pada tahap analisa kebuthan sistem, dilakukan analisa terhadap apa saja kebutuhan dari pembangunan sistem yang akan dilakukan. Analisanya meliputi analisa kebutuhan alat dan bahan yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai berikut:

- a. Kebutuhan Software :
 1. Fitur Menghidupkan dan Mematikan AC
 2. Fitur Input Jadwal Matakuliah
 3. Arduino IDE versi 1.8.12 sebagai *code editor* untuk mikrokontroler
 4. Android Studio versi 3.6.1 sebagai *code editor*
- b. Kebutuhan Hardware :
 1. PC/ Laptop yang akan digunakan untuk membangun aplikasi dan untuk menguji sistem nantinya.

2. Sistem Operasi yang digunakan adalah Windows 10
3. *Smartphone* Android untuk menguji coba sistem
4. AC LG yg sudah terpasang di ruang kelas A3-02 PSTI UNRAM
5. 1 buah Wemos D1 (R2) sebagai mikrokontroler
6. 1 buah sensor PIR sebagai pendeteksi keberadaan seseorang
7. 1 buah IR Transmitter sebagai transmisi sinyal code ke AC
8. 1 buah IR Receiver untuk merekam IR code dari remote AC
9. 1 buah breadboard
10. 1 set kabel jumper
11. Remote AC

3.3. Rancangan Arsitektur Sistem

Pada Gambar 3.2 merupakan gambaran arsitektur sistem yang akan dibuat



Gambar 3.2 Rancangan Arsitektur Sistem

Berikut merupakan penjelasan dari masing-masing proses beserta hubungan antar proses yang terdapat pada Gambar 3.2 :

1. IR *Receiver* (A) pada rangkaian elektronik digunakan untuk mengambil IR code dari remote AC menggunakan IR *Receiver* diterima oleh mikrokontroler yang akan dijadikan sebagai kode perintah untuk mengontrol AC.
2. Sensor (B) adalah sensor PIR yang digunakan untuk mendeteksi gerakan manusia. Hasil dari pembacaan sensor diterima oleh mikrokontroler untuk melakukan

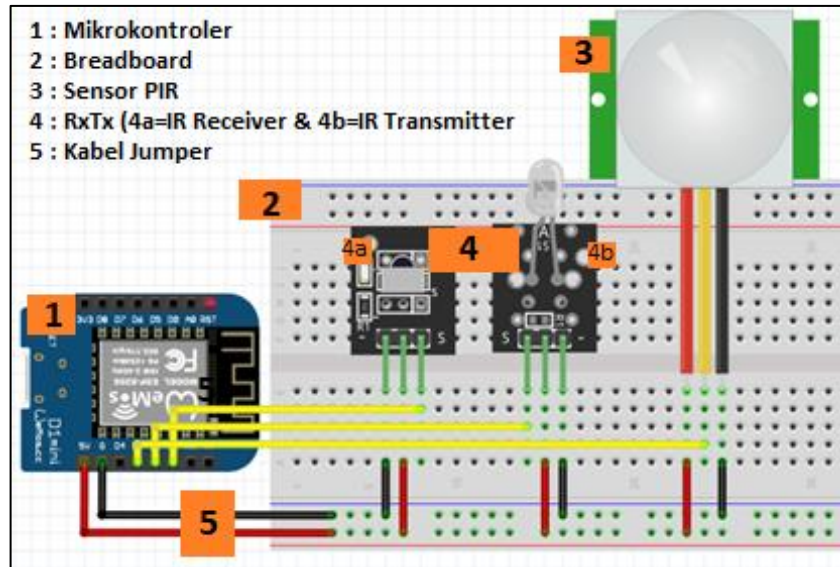
pengolahan dalam menentukan kondisi yang akan dikirimkan ke AC (G) menggunakan IR Transmitter.

3. IR Transmitter(C) pada rangkaian elektronik digunakan untuk mengirim sinyal perintah ke AC berdasarkan instruksi yang telah diolah oleh mikrokontroler.
4. Mikrokontroler Wemos D1 (R2) digunakan untuk mengambil data dari sensor, mengolah data dan melakukan perintah ke AC dengan IR *Transmitter* serta memiliki beban *library* ESP8266WiFi untuk dapat terkoneksi dengan internet, *library* PubSubClient yang digunakan untuk dapat terhubung dengan server broker sebagai *client* dan juga *library* ESP8266 HTTP *Client* yang digunakan untuk dapat mengakses layanan web. Mikrokontroler mengirim nilai sensor yang didapat menuju server broker.
5. Server Broker (E) merupakan server yang akan dituju oleh *client* dalam penyampaian data dimana pengiriman datanya menggunakan protokol MQTT(F).
6. *Smartphone* Android (D) perangkat yang digunakan untuk mengontrol sistem dan menambahkan jadwal matakuliah yang akan digunakan.
7. Database (E) digunakan untuk menyimpan jadwal matakuliah sekaligus sebagai tempat penyimpanan data.
8. *Air Conditioner* (G) perangkat elektronik yang akan dikendalikan oleh sistem.

Pengambilan IR *code* dari *remote* AC akan dilakukan pada saat pengujian yang kemudian akan digunakan sebagai perintah kontrol yang dikirimkan oleh mikrokontroler Wemos D1 melalui IR *Transmitter* menuju AC.

3.4. 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 3.3 Rancangan Perangkat Keras

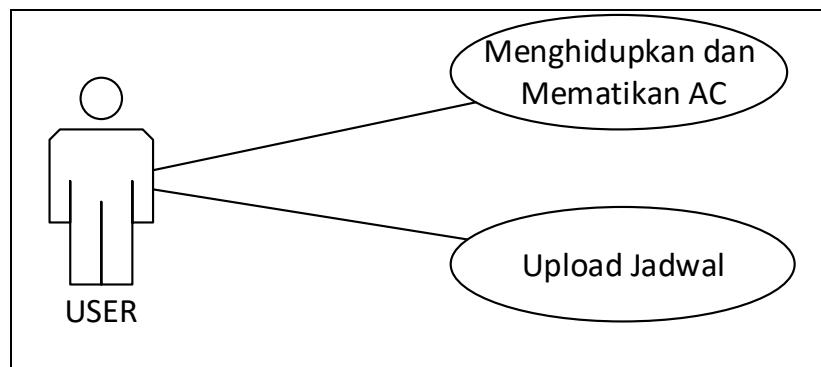
3.5. Rancang Perangkat Lunak

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

3.5.1. Desain Sistem

Berikut merupakan rancangan dari desain sistem yang akan dibuat :

1. *Usecase Diagram*

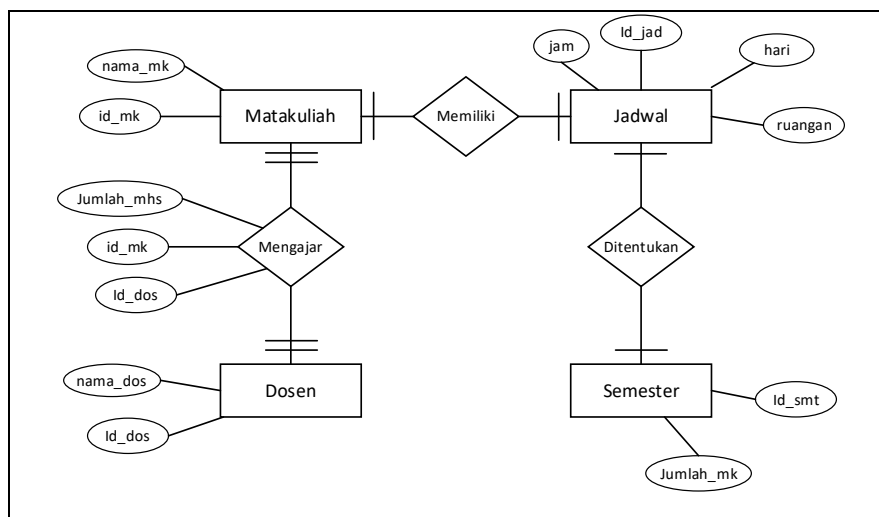


Gambar 3.4 Rancangan *Usecase Diagram*

Pada Gambar 3.4 merupakan rancangan *Usecase diagram* pada sistem yang dibuat. Ada 1 aktor yaitu *user* yang merupakan Staf PSTI UNRAM. *User* memiliki kendali dapat melakukan semua aktifitas yang ada yaitu menghidup-matikan AC, mengupload jadwal matakuliah.

2. *Entity Relationship Diagram (ERD)*

Pada Gambar 3.5 merupakan rancangan ERD dari sistem. Terdapat beberapa entitas yaitu Matakuliah, Jadwal, Dosen dan Semester yang saling terhubung dengan relasinya masing-masing.



Gambar 3.5 Rancangan *Entity Relationship Diagram*

Berdasarkan ERD pada Gambar 3.5 maka didapatkan tabel *database* sebagai berikut :

a. Tabel Matakuliah

Tabel 3.1 *Database* Matakuliah

No	Nama	Tipe data	Null	Default	Keterangan
1	Id_mk	Int(4)	No	None	Primary key
2	Nama_mk	varchar(25)	No	None	

b. Tabel Dosen

Tabel 3.2 *Database* Dosen

No	Nama	Tipe data	Null	Default	Keterangan
1	Id_dos	int(4)	No	None	Primary key
2	Nama_dos	varchar(70)	No	None	

c. Tabel Jadwal

Tabel 3.3 *Database* Jadwal

No	Nama	Tipe data	Null	Default	Keterangan
1	Id_jad	Int(4)	No	None	Primary key
2	Hari	varchar(6)	No	None	
3	Ruangan	varchar(5)	No	None	
4	Jam	varchar(6)	No	None	

d. Tabel Semester

Tabel 3.4 *Database* Semester

No	Nama	Tipe data	Null	Default	Keterangan
1	Id_smt	int(2)	No	None	Primary key
2	Jumlah_mk	int(2)	No	None	

e. Tabel Mengajar

Tabel 3.5 Database Mengajar

No	Nama	Tipe data	Null	Default	Keterangan
1	Id_mk	int(4)	No	None	Primary key
2	Id_dos	int(4)	No	None	Foreign key
3	Jumlah_mhs	int(2)	No	None	

3. Desain Interface

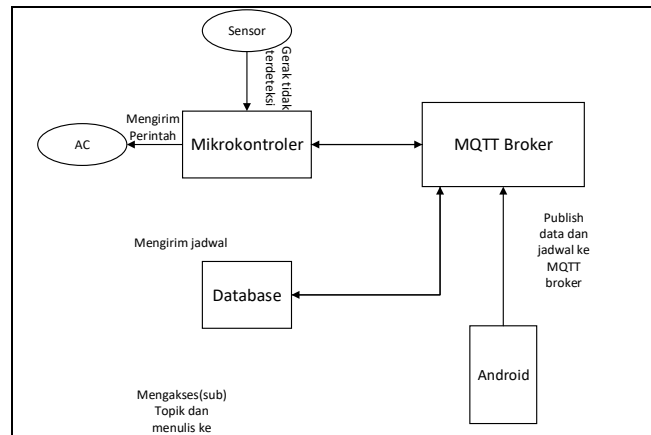
Pada Gambar 3.6 merupakan desain beranda untuk antar muka sistem yang akan dibuat. Pada halaman beranda terdapat tombol untuk menghidupkan dan mematikan AC dan juga terdapat tombol untuk meng-upload jadwal.



Gambar 3.6 Rancangan Beranda dan Halaman Upload Jadwal Aplikasi Android

3.5.2. 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 3.7 Rancangan Komunikasi MQTT

3.6. 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. Tahap Implementasi akan dibagi menjadi 2 proses yaitu:

1. Penyusunan perangkat keras

Pada tahap ini, mikrokontroler Wemos D1, sensor PIR, IR *Transmitter*. Akan disusun menjadi satu rangkaian perangkat keras yang nantinya akan digunakan untuk mengontrol AC secara otomatis. Proses penyusunan perangkat disesuaikan dengan rancangan yang sudah dibuat pada tahap rancangan perangkat keras sistem.

2. Pembuatan perangkat lunak Pada tahapan ini akan dilakukan pembangunan sistem berbasis Android dengan menggunakan *software* Android Studio. Aplikasi Android yang dibangun akan digunakan untuk mengontrol penggunaan sistem dalam penggunaan AC sesuai jadwal matakuliah.

3.7. Pengujian dan Evaluasi Sistem

Pada tahap pengujian dan evaluasi sistem nantinya akan dilakukan pengujian terhadap sistem yang telah dibangun dan mengevaluasi hasil dari pengujian yang telah dilakukan. Ada 2 tahapan yaitu:

1. Pada tahap pertama dilakukan uji coba terhadap fungsi dari perangkat yang telah dibuat untuk memastikan komponen-komponen pada perangkat keras berfungsi sebagaimana mestinya yang kemudian menguji sistem yang akan terhubung dengan protokol MQTT untuk dapat mengontrol AC secara *real time*.

2. Tahap berikutnya adalah menguji kelayakan fungsi fitur dari aplikasi Android yang sudah dibuat. Diantaranya dapat mengontrol AC (hidupkan dan matikan) dan dapat mengunggah jadwal matakuliah.

Sistem akan diuji pada ruang kelas A3-02 dengan menggunakan beberapa mahasiswa untuk melakukan percobaan pada ruangan kelas dan pengujian oleh staf prodi. 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.

3.8. 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.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Realisasi Sistem

Bab ini akan membahas tentang hasil dari penelitian yang dilakukan sesuai dengan judul penelitian yaitu “Sistem Penjadwalan AC Ruangan 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.

4.1.1 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 4.1 Realisasi Perangkat Keras

Pada Gambar 4.1, 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. Fungsi dari masing-masing perangkat keras tersebut adalah sebagai berikut:

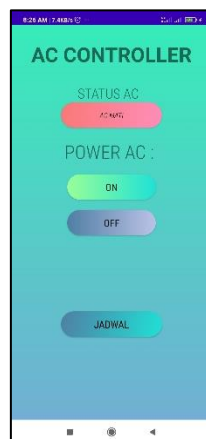
1. *Wemos D1*, Adalah bagian inti yang mengontrol modul yang terpasang pada rangkaian perangkat keras sistem dan digunakan sebagai penghubung ke jaringan internet karena mikrokontroler *wemos D1* ini memiliki modul ESP8266.

2. IR *Transmitter*, digunakan untuk mengirim sinyal inframerah yang berfungsi sebagai perintah ke AC berdasarkan instruksi yang telah diproses oleh mikrokontroler apakah akan menghidupkan atau mematikan AC.
3. IR *Receiver* berfungsi untuk merekam kode dari remot AC yang digunakan untuk nantinya sebagai instruksi yang dikirimkan melalui IR *Transmitter*.
4. Sensor PIR berfungsi untuk mendeteksi gerakan dengan mendeteksi pancaran sinar inframerah yang dihasilkan oleh manusia.

4.1.2 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 4.2 Tampilan Menu Awal Aplikasi

Gambar 4.2 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.

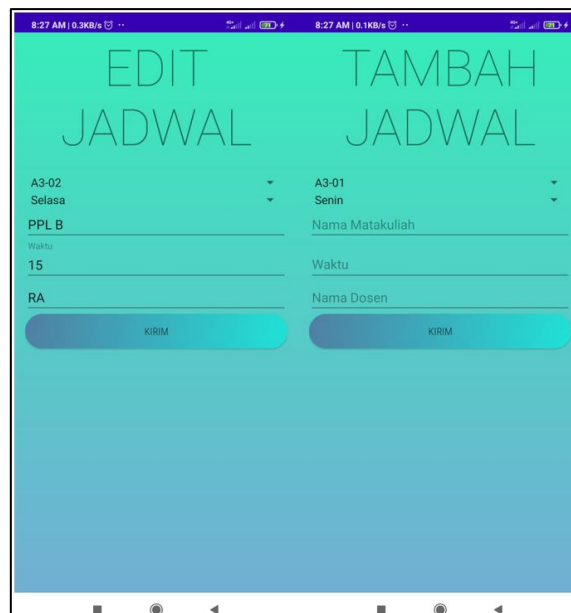
Tampilan Menu Jadwal :



Gambar 4.3 Tampilan Menu Jadwal

Gambar 4.3 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 4.4 Tampilan Menu *Form* Tambah dan Edit Jadwal Aplikasi

Gambar 4.4 diatas merupakan tampilan edit dan tambah jadwal dari aplikasi android. Terdapat beberapa kolom yang memiliki fungsi berbeda. Diantaranya adalah kolom Ruangn 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.

4.1.3 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-inputnya 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. import context
2. import paho.mqtt.client as mqtt
3. import mysql.connector
4. import paho.mqtt.publish as publish
5. import datetime
```

Kode Sumber 4.1 *Script* Modul/Package

Script diatas merupakan modul yang digunakan dalam pembuatan *script*nya. Fungsi dari masing-masing modul yaitu “context” untuk memastikan Paho(MQTT broker) ada di PythonPATH. Kemudian “paho.mqtt.client as mqtt” untuk memanggil fungsi modul Paho MQTT *Client* sebagai “mqtt” yang akan digunakan sebagai penghubung ke broker, beda halnya dengan “paho.mqtt.publish as publish” yang berfungsi untuk memanggil fungsi *publish* sebagai “publish” agar bisa digunakan secara terpisah, lalu “mysql.connector” berfungsi untuk memanggil fungsi MySQL ke dalam *script* dan “datetime” berfungsi untuk menambahkan fungsi waktu pada *script*.

```
1. mydb = mysql.connector.connect(host="***",
2. user="*****",
3. password="***",
4. database="dbac")
```

Kode Sumber 4.2 *Script* Deklarasi Database

Script diatas merupakan deklarasi dari *database* yang digunakan, dimana *datasenya* menggunakan *localhost* dan *user default* dengan tidak menggunakan *password*, nama *datasenya* adalah “dbac”.


```

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("alamat.broker.com", port)
6. mqtta.subscribe([("jmcdh",1), ("jmcdr",1), ("jmcd000",1), ("jmcd001",1), ("jmcd002",1)])

```

Kode Sumber 4.3 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("alamat.broker.com",port)" bermaksud melakukan koneksi kepada "broker.hivemq.com" pada port "port" . 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.
8.     if hari and ruang and namat and jam and namdos != None:
9.         mycursor = mydb.cursor()
10.         sql = "INSERT INTO jadwal (Hari,Ruangan,Namat,Jam,Namdos) VALUES (%s,%s,%s,%s,%s) ON DUPLICATE KEY UPDATE Namat=%s"
11.         val = (hari,ruang,namat,jam,namd, namat)
12.         mycursor.execute(sql,val)
13.         mydb.commit()
14.         print(mycursor.rowcount, "record inserted.")

```

Kode Sumber 4.4 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 on_har(x):
2.     global hari
3.     if x.topic=="jmcdh":
4.         hari=str(x.payload.decode("utf-8"))

```

```

5.         return hari
6. def on_rua(x):
7.     global ruang
8.     if x.topic=="jmcdr":
9.         ruang=str(x.payload.decode("utf-8"))
10.        return ruang
11.    def on_nama(x):
12.        global namat
13.        if x.topic=="jmcd000":
14.            namat=str(x.payload.decode("utf-8"))
15.            return namat
16.    def on_ja(x):
17.        global jam
18.        if x.topic=="jmcd001":
19.            jam=str(x.payload.decode("utf-8"))
20.            return jam
21.    def on_namd(x):
22.        global namdos
23.        if x.topic=="jmcd002":
24.            namdos=str(x.payload.decode("utf-8"))
25.            return namdos

```

Kode Sumber 4.5 *Script* Deklarasi Fungsi Mengambil Pesan Melalui MQTT

Script diatas merupakan pembentukan fungsi untuk mengambil pesan yang terkirim sesuai dengan topik yang telah *unsubscribe* pada MQTT broker. Adapun *script* “on_har” untuk mengambil pesan hari dengan topik “jmcdh”, *script* “on_rua” untuk mengambil pesan ruangan dengan topik “jmcdr”, *script* “on_nama” untuk mengambil pesan mata kuliah dengan topik “jmcd000”, *script* “on_ja” untuk mengambil pesan waktu dengan topik “jmcd001”, dan *script* “on_namd” untuk mengambil pesan dosen dengan topik “jmcd002”.

```

1. def getpub():
2.     global jam1,jam2,jam3,jam4,day1,day2,day3,day4,day5,menit1,m
   enit2,menitis,menitpul,jampul,jamis
3.     jam1="7"
4.     jam2="10"
5.     jam3="13"
6.     jam4="15"
7.     jamis="12"
8.     jampul="16"
9.     day1="1"
10.    day2="2"
11.    day3="3"
12.    day4="4"
13.    day5="5"
14.    menit1="30"
15.    menit2="10"
16.    menitis="40"
17.    menitpul="10"
18.    x = datetime.datetime.now()
19.    y=x.strftime("%w")
20.    print(y)
21.    z=x.strftime("%H")
22.    print(z)

```

```

23.         v=x.strftime ("%M")
24.         print (v)
25.         t=x.strftime ('%Y-%m-%d %H:%M:%S')
26.         print (t)

```

Kode Sumber 4.6 *Script* Deklarasi Fungsi Untuk Mengambil Data Dari *Database*

Script diatas merupakan pembuatan fungsi “getpub()” untuk mengirimkan pesan menghidupkan atau mematikan AC sesuai dengan jadwal matakuliah yang telah *diinput* ke dalam *database* MySQL. Penggalan *Script* di dalam fungsi “getpub()” adalah variabel-variabel pendukung untuk mengkondisikan hari dan waktu sesuai dengan saat ini atau *realtime*. Dengan memanfaatkan hari dan jam masuk dari setiap mata kuliah, dimana “day1” adalah hari senin kemudian “day2” adalah hari selasa dan seterusnya. Kemudian “jam1” adalah jam pertama atau jam 7:30 yg disesuaikan dengan “menit1” dan “menit2” sebagai menitnya, dimana “menit1” adala 30 menit dan “menit2” adalah 10 menit. *Script* “x = datetime.datetime.now()” untuk membuat variabel “x” sebagai wadah untuk waktu saat ini, *script* “y=x.strftime (“%w”)” untuk membuat variabel “y” menyimpan hari sebagai nomor dimana 1= senin, 2= selasa, dst. , *script* “z=x.strftime (“%H”)” untuk membuat variabel “z” untuk menyimpan waktu saat ini dalam bentuk jam saja sedangkan *script* “v=x.strftime (“%M”)” untuk menyimpan waktu saat ini dalam bentuk menit saja pada variabel “v” dan yang terakhir ada “t=x.strftime (“%Y-%m-%d %H:%M:%S”)” yang berfungsi untuk menjadikan variabel “t” sebagai wadah untuk satuan waktu “Tahun-Bulan-Tanggal Jam:Menit:Detik” untuk waktu saat ini.

```

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

```

Kode Sumber 4.7 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 Stat
   us`)VALUES (%s,%s)"
15.            val = (t,e)
16.            mycurso.execute(sql,val)
17.            mydb.commit()
18.            print(mycurso.rowcount, "Data Diterima, AC T
   elah Dihidupkan.")
19.        else:
20.            print("Jam Kosong")
```

Kode Sumber 4.8 *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 diabil tidak kosong yang ditandai dengan “mycurso.rowcount==1” untuk menandai kolom yang terambil akan bernilai “1” jika ada. 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”) jika sudah pernah dikirim maka *script* hanya akan menampilkan tulisan “AC sedang menyala” jika belum pernah dikirim MQTT pada hari dan jam saat ini maka dilakukan proses berikutnya yaitu menjalankan *script* “publish.single ("onindong", v, hostname="broker.hivemq.com")” akan berjalan dimana *script* tersebut berfungsi untuk melakukan *publish* pesan MQTT yang memiliki topik “onindong” dengan pesan pada variabel “v” yang memiliki nilai “1”

dan dikirim ke broker “broker.hivemq.com”. Setelah melakukan *publish* selanjutnya *script* akan menulis laporan AC hidup pada jam saat ini dengan menuliskan jam dan status pada *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”. Namun jika kondisi pertama terpenuhi namun kondisi kedua tidak terpenuhi yaitu jika data berhasil diambil dari *database*, maka akan menampilkan tulisan “Jam Kosong”. Begitupun kondisi pada hari-hari berikutnya dari hari senin sampai jumat yang tersimpan pada variabel “day2, day3, day4 dan day5” sampai di kondisi hari terakhir “elif y==day5 and z==jam4 and v>=menit2”.

```

1. elif y==day1 and z==jamis and v>=menitis:
2.     v =0
3.     publish.single("offindong", v, hostname="alamat.bro
ker.com")
4.     mycursorr = mydb.cursor()
5.     e="Mati"
6.     sql = "INSERT INTO log(`Log Waktu`,`Log Status`)VAL
UES(%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 4.9 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 menjalankan *script* “publish.single("offindong", v, hostname="alamat.broker.com)” untuk mengirim pesan “0” yang sudah tersimpan pada variabel “v” dengan topik “offindong” dan melalui broker host “alamat.broker.com” . Setelah itu akan dilakukan pencatatan laporan AC mati pada jam tersebut dengan *syntax query* “INSERT INTO log(`Log Waktu`, `Log Status`) VALUES (%s , %s)” ke dalam *database*. 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.

4.1.4 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. WiFiClient espClient;
2. PubSubClient client(espClient);
3. int value = 0;
4. int pirPin = D7;
5. int val;
6. int stat=1;
7. const uint16_t kIrLed = D2;
8. IRsend irsend(kIrLed);
9. uint16_t ondong[139] = {8986, 4498, 656, 1682, 656, 550, 652,
556, 650, 1684, 652, 554, 656, 552, 652, 1684, 654, 552,
654, 552, 652, 552, 656, 1682, 652, 556, 650, 556, 652, 554, 650,
556, 650, 558, 652, 554, 652, 556, 654, 554, 650, 558, 652, 554,
654, 1682, 652, 554, 656, 550, 652, 554, 630, 578, 628, 578, 650,
558, 650, 1684, 628, 580, 624, 1710, 624, 586, 652, 554, 650, 1686,
650, 554, 654, 20684, 652, 1686, 650, 556, 648, 560, 678, 526,
654, 1684, 656, 552, 650, 554, 656, 550, 680, 526, 652, 554,
652, 552, 656, 550, 654, 552, 626, 1712, 648, 556, 658, 552, 654,
552, 656, 550, 656, 550, 654, 554, 650, 556, 654, 552, 656, 550,
650, 554, 628, 578, 650, 560, 652, 552, 656, 550, 656, 550, 628,
1710, 654, 492, 716, 1682, 650};
10. uint16_t offdong[139] = {8982, 4500, 650, 1688, 648, 556,
624, 582, 626, 580, 652, 558, 648, 558, 650, 556, 650, 556, 650,
556, 654, 552, 646, 560, 626, 580, 650, 556, 626, 582, 624, 586,
648, 558, 648, 556, 648, 560, 650, 556, 650, 556, 650, 558, 622,
1714, 646, 558, 650, 558, 624, 582, 648, 560, 622, 586, 650, 556,
650, 1686, 650, 556, 650, 1686, 650, 558, 648, 556, 650, 1688,
650, 556, 650,
11. 20692, 650, 558, 648, 1686, 652, 556, 650, 558, 652, 1684,
654, 552, 650, 558, 650, 554, 654, 492, 708, 560, 648, 560, 650,
504, 704, 554, 652, 1686, 626, 578, 650, 556, 654, 492, 686,
582, 650, 556, 652, 554, 648, 558, 652, 554, 634, 576, 648, 558,
652, 554, 648, 558, 652, 556, 652, 554, 654, 554, 652, 1684, 648,
1688, 654, 1684, 652};

```

Kode Sumber 4.10 Deklarasi Variabel Mikrokontroler

Script diatas merupakan deklarasi variabel dari *library* yang digunakan dan deklarasi variabel tambahan pada program. Adapun *script* “WiFiClient espClient” berfungsi mendefinisikan *library* “ESP8266WiFi.h” kemudian *script* “PubSubClient client(espClient)” mendefinisikan *library* “PubSubClient” dan variabel “IRsend irsend(kIrLed)” mendefinisikan *library* “IRsend”. *Script* “int pirPin = D7” berfungsi untuk mendefinisikan PIN modul untuk PIR sensor dan *script* “const uint16_t kIrLed = D2” untuk PIN modul IR *transmitter*. Adapun *script* “uint16_t ondong[139]” dan “uint16_t offdong[139]” untuk mendefinisikan RAW data yang telah diambil dengan IR *receiver* dan remote AC.

RAW AC data yang digunakan adalah AC SHARP sesuai dengan yang terpasang pada kelas A3-02.

```
1. void setup_wifi() {
2.   delay(10);
3.   Serial.println();
4.   Serial.print("Connecting to ");
5.   Serial.println(ssid);
6.   WiFi.mode(WIFI_STA);
7.   WiFi.begin(ssid, password);
8.   while (WiFi.status() != WL_CONNECTED) {
9.     delay(500);
10.    Serial.println("WiFi connected");
11.    Serial.println("IP address: ");
12.    Serial.println(WiFi.localIP());}
```

Kode Sumber 4.11 Fungsi Wifi Pada Program Mikrokontroler

Script diatas merupakan fungsi untuk melakukan koneksi wifi pada jaringan wifi yang sudah ditentukan pada SSID dan *password*-nya pada deklarasi variabel sebelumnya. Jadi mikrokontroler akan otomatis tersambung dengan jaringan wifi tersebut apabila masih dalam jangkauan mikrokontroler.

```
1. void reconnect() {
2.   while (!client.connected()) {
3.     Serial.print("Attempting MQTT connection...");
4.     String clientId = "ESP8266Client-";
5.     clientId += String(random(0xffff), HEX);
6.     if (client.connect(clientId.c_str())) {
7.       Serial.println("connected");
8.       client.subscribe("onindong");
9.       client.subscribe("offindong");
10.    } else {
11.      Serial.print("failed, rc=");
12.      Serial.print(client.state());
13.      Serial.println(" try again in 5 seconds");
14.      delay(5000); } }}
```

Kode Sumber 4.12 Fungsi Untuk Mengkoneksikan Kembali Pada Mikrokontroler

Script diatas merupakan fungsi untuk menghubungkan mikrokontroler ke jaringan wifi jika sewaktu-waktu terputus, juga berfungsi sebagai fungsi *subscribe topic* MQTT . Fungsi ini akan terus berjalan selama mikrokontroler belum terkoneksi ke jaringan wifi dan akan terus mencoba terhubung dengan interpal waktu 5 detik.

```
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.     Serial.print((char)payload[i]);
7.   }
8.   Serial.println();
9.   if ((char)payload[0] == '1') {
10.    irsend.sendRaw(ondong, 139, 38);
```

```

10.     digitalWrite(BUILTIN_LED, LOW);  }
11.     else {
12.         irsend.sendRaw(offdong, 139, 38);
13.         digitalWrite(BUILTIN_LED, HIGH);
14.         stat=0;}}

```

Kode Sumber 4.13 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 “onindong” untuk menerima pesan “1” yang berfungsi menghidupkan AC dan topik “offindong” 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 setup() {
2.     Serial.begin(115200);
3.     setup_wifi();
4.     client.setServer(mqtt_server, port);
5.     client.setCallback(callback);
6. }

```

Kode Sumber 4.14 Fungsi *Setup* Program Mikrokontroler

Script diatas merupakan fungsi yang akan berjalan pertama kali pada saat program pertama kali dijalankan. Adapun yang dilakukan antara lain “`Serial.begin(115200);`” berfungsi menjalankan mikrokontroler pada kecepatan “115200”, kemudian “`setup_wifi();`” berfungsi menjalankan fungsi wifi untuk menyambungkan ke jaringan wifi yang sudah terdaftar. *Script* “`client.setServer(mqtt_server, port)`” berfungsi untuk melakukan koneksi ke broker yang sudah dideklarasikan sebelumnya dan “`client.setCallback(callback)`” berfungsi melakukan *subscribe* pada topik yang sudah tertulis dalam fungsi tersebut.

```

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(val==LOW){
14.         if (stat==1 and now - lastMsg > 1800000) {
15.             lastMsg = now;
16.             irsend.sendRaw(offdong, 139, 38);
17.             stat=0;
18.             digitalWrite(BUILTIN_LED, HIGH);
19.         }
20.         else if(stat==0 and now - lastMsg > 1800000){
21.             Serial.print("AC Dimatikan");
22.         }}
23.     else{
24.         lastMsg = now;}}

```

Kode Sumber 4.15 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(val==LOW)*” dan “*if (stat==1 and now - lastMsg > 1800000) {*” fungsi ini akan mematikan AC jika “*if(val==LOW)*” Ruangan sedang tidak ada orang di dalam ruangan, lalu *script* “*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, 139, 38);*” kemudian membuat status AC menjadi mati “*stat=0*”.

4.2 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.

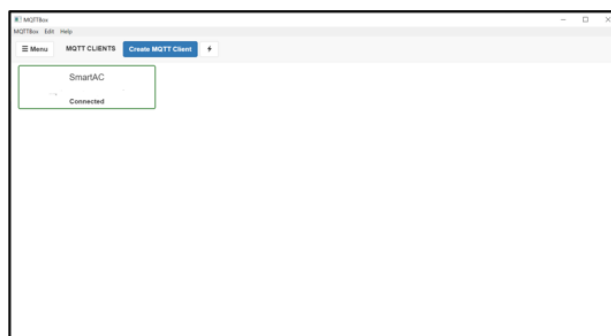
4.2.1 Hasil Pengujian Sistem Aplikasi Android

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 4.1 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	Melakukan 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

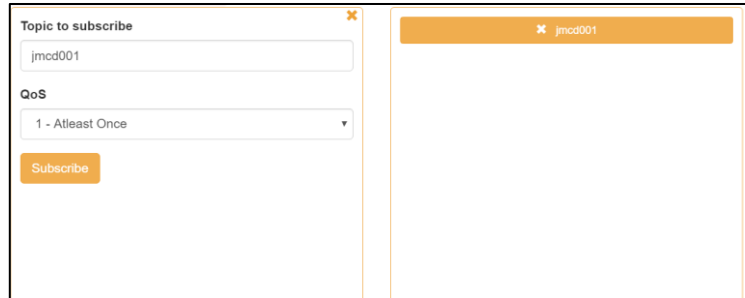
Pengujian dilakukan dengan bantuan aplikasi MQTTBox untuk menerima pesan MQTT yang dikirim dari android ke broker dengan topik tertentu. Berikut tampilan awal dari MQTTBox.



Gambar 4.5 Tampilan Awal MQTT Box

Pada Gambar 4.4 MQTTBox sudah terhubung dengan broker hive/Paho dengan alamat “alamat.broker.com”. MQTTBox melakukan *subscribe* terhadap topik-topik yang digunakan oleh aplikasi android diantaranya adalah “onindong” untuk mengirim pesan 1 , “offindong” untuk mengirim pesan 0. Kedua topik itu berada dalam fungsi tombol ON OFF pada menu home di aplikasi android seperti pada gambar 4.2. Topik

lain yang di-*subscribe* oleh MQTTBox adalah “jmcdh” untuk mengirim pesan hari (Senin,Selasa,dst..), ada juga topik ruangan “jmcdr” untuk ruangan yang telah dibuat statis bersama dengan hari. Selain itu ada juga topik “jmcd000” untuk matakuliah, “jmcd001” untuk jam dan “jmcd002” untuk kode dosen. Berikut tampilan MQTTBox yang telah mensubscribe topik.

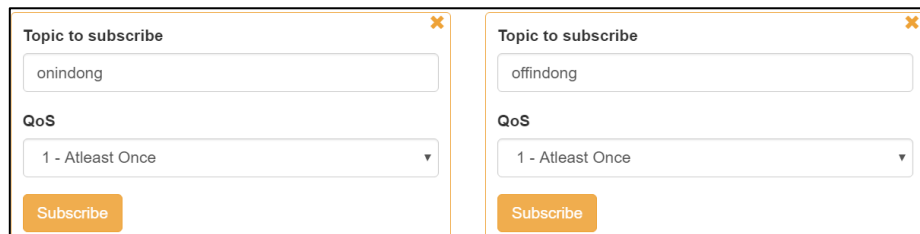


Gambar 4.6 Contoh MQTTBox Sebagai *Subscriber*

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

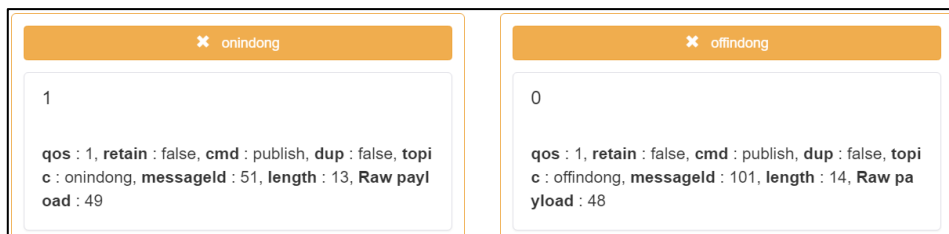
1. Fungsi ON OFF

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



Gambar 4.7 *Subscribe* Untuk Tombol ON dan OFF

Pada Gambar 4.7 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 4.8 Hasil Pesan Yang Diterima *Subscribe* ON dan OFF

Pada Gambar 4.8 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 Kirim Jadwal

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

Gambar 4.9 *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 4. 10 Hasil Pesan Yang Diterima *Subscribe* Form *Upload*

Pada Gambar 4.10 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 “jmc000”, untuk data Jam mengirimkan pesan “10:10” dengan topik “jmc001” dan untuk data Kode Dosen mengirimkan pesan “OKE” dengan topik “jmc002”.

4.2.2 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 4.11.



Gambar 4.12 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 makan akan menampilkan pesan seperti pada gambar 4.12 dan tidak mengirim pesan MQTT apapun.

```

C:\WINDOWS\system32>"C:\Users\jackmalik79\AppData\Local\Microsoft\Windows\apps\examples\client_sub.py"
Belum Ada Kelas
rc: 0
Subscribed: None

```

Gambar 4.12 *Task Scheduler* Saat Tidak Ada Jadwal

Namun jika ada jadwal yang didapatkan di dalam *database* maka akan menampilkan pesan seperti pada gambar 4.12 , apabila program sudah berhasil dijalankan pada jam yang sama sebelumnya maka akan mengirim pesan untuk menghidupkan/mematikan AC, lalu menampilkan pesan seperti pada gambar 4.13.

```

C:\Windows\System32\cmd.exe - python die
Microsoft Windows [Version 10.0.18363.12]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\paho.mqtt.python-master\paho.mqtt.pyt
1 Data Diterima, AC Telah Dihidupkan.
rc: 0
Subscribed: None

```

Gambar 4.13 *Task Scheduler* Saat Ada Jadwal Pada Waktu Program Dijalankan

```

E:\paho.mqtt.python-master\pa
AC Sedang Menyala
rc: 0
Subscribed: None

```

Gambar 4.14 *Task Scheduler* Saat AC Sudah Pernah Dijalankan Di Jam Yang Sama. Salah satu hasil pengujian yang didapatkan dari pengujian TS yaitu pada hari jumat 18 desember 2020 dapat dilihat pada gambar 4.14.

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 4.15 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 4.15

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

Gambar 4.16 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 4.16.

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		

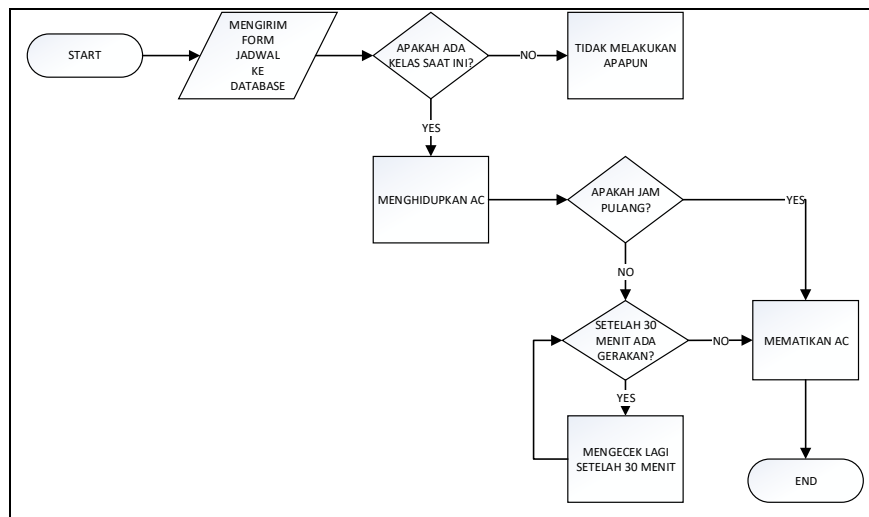
Gambar 4.17 Hasil Log Uji *Task Scheduler* Pada Jadwal

Pada Gambar 4.17 menggunakan jadwal yang berlangsung pada hari selasa dan rabu yang terdaftar pada jadwal yang berlaku. Adapun jadwal yang digunakan dapat dilihat pada gambar 4.18

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 4.18 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. Untuk lebih jelasnya dapat dilihat pada *Flowchart* Gambar 4.18.



Gambar 4.119 *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.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian dan juga pengujian yang telah dilakukan, 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 ruangan 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 beberaa 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.

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%20Sistem%20Monitoring%20Suhu%20Jarak%20Jauh%20Berbasis%20Internet%20of%20Things%20Menggunakan%20Protokol%20MQTT.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 Periferal 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 Ruangan Menggunakan Sensor Passive Infra Red (PIR) Dilengkapi Kontrol Penerangan Pada Ruangan 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 Phasa," *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 Penyus 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>.

Bukti ACC Online :

