

**ADAPTIVE CLASSROOM BERBASIS IOT (INTERNET OF
THINGS), MANAJEMEN PENGGUNAAN AIR CONDITIONER (AC)
SECARA OTOMATIS**

Tugas Akhir

Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh :

Muhamad Pahrurrozi

F1D 015 051

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2020

TUGAS AKHIR

ADAPTIVE CLASSROOM BERBASIS IOT (INTERNET OF THINGS), MANAJEMEN PENGGUNAAN AIR CONDITIONER (AC) SECARA OTOMATIS

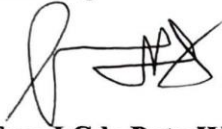
Oleh

MUHAMAD PAHRURROZI

F1D 015 051

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama



Tanggal : 2/03/2020

Dr.Eng. I Gde Putu Wirarama W W, S.T., M.T.

NIP. 19840919 201803 1 001

2. Pembimbing Pendamping



Tanggal : 2/03/2020

Arivan Zubaidi, S.Kom., M.T.

NIP: 19860913 201504 1 001

Mengetahui
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Prof. Dr. Eng. I Gede Pasek Suta Wijaya, S.T., M.T.

NIP: 19731130 200003 1 001

TUGAS AKHIR
ADAPTIVE CLASSROOM BERBASIS IOT (INTERNET OF THINGS), MANAJEMEN PENGGUNAAN AIR CONDITIONER (AC) SECARA OTOMATIS

Oleh :

MUHAMAD PAHRURROZI
F1D 015 051

Telah dipertahankan di depan Dewan Penguji
Pada tanggal 28 Februari 2020
dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1
Program Studi Teknik Informatika

Susunan Tim Penguji

1. Penguji 1



Tanggal : 2 - 3 - 2020

Andy Hidayat Jatmika, S.T., M.Kom.
NIP. 19831209 201212 1 001

2. Penguji 2



Tanggal : 02 - 03 - 2020

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

3. Penguji 3



Tanggal : 06 - 03 - 2020

Arik Aranta, S.Kom., M.Kom.
NIP. 19920323 201903 1 012

Mataram, 6 Maret 2020

Dekan Fakultas Teknik
Universitas Mataram



Akmaluddin, S.T., M.Sc.(Eng.), Ph.D.
NIP. 19681231 199412 1 001

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya yang bertanda tangan di bawah ini bahwa dalam Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Mataram, 28 Februari 2020

Muhamad Pahrurrozi

PRAKATA

Puji syukur penulis panjatkan kepada Allah SWT Maha Pengasih dan Penyayang atas segala berkat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir dengan judul “*Adaptive Classroom* Berbasis IoT (*Internet of Things*), Manajemen Penggunaan *Air Conditioner* (AC) Secara Otomatis”.

Tugas Akhir ini dilaksanakan di Program Studi Teknik Informatika Fakultas Teknik Universitas Mataram (PSTI FT UNRAM). Tujuan dari Tugas Akhir ini adalah untuk membangun suatu sistem yang dapat digunakan oleh PSTI FT UNRAM untuk manajemen penggunaan AC secara otomatis pada setiap ruang kelas PSTI FT UNRAM. Sistem yang dibangun sekaligus dapat dilakukan pemantauan dari sistem berbasis *web*.

Dalam penulisan Tugas Akhir ini tentunya tidak luput dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis. Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna sehingga penulis membutuhkan kritik dan saran yang membangun untuk kemajuan teknologi di masa yang akan datang.

Akhir kata semoga tidaklah terlampau berlebihan, bila penulis berharap agar karya ini dapat bermanfaat bagi pembaca.

Mataram, 28 Februari 2020

Penulis

UCAPAN TERIMA KASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materil dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada :

1. Keluarga yang selama ini telah memberikan doa dan dukungannya dalam menjalani perkuliahan dan pengerjaan Tugas Akhir ini sehingga dapat terselesaikan.
2. Bapak Dr. Eng. I Gde Putu Wirarama Wedashwara W., S.T., M.T. selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
3. Bapak Ariyan Zubaidi, S.Kom., M.T. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini.
4. Dosen penguji, atas pemberian kritik dan saran yang bersifat membangun yang sangat diperlukan dalam penyelesaian Tugas Akhir ini.
5. Mahasiswa Teknik Informatika yang telah memberikan dukungan dan semangat, khususnya mahasiswa angkatan 2015.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan dan bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini.

Semoga Allah SWT memberikan imbalan yang setimpal atas bantuan yang telah diberikan kepada penulis.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN TIM PEMBIMBING	ii
HALAMAN PENGESAHAN TIM PENGUJI.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PRAKATA.....	v
UCAPAN TERIMA KASIH	vi
DAFTAR ISI	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
DAFTAR LISTING	xi
ABSTRAK	xii
ABSTRACT.....	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1. Tinjauan Pustaka	5
2.2. Dasar Teori.....	7
2.2.1 Nyaman Termal	7
2.2.2 <i>Air Conditioner (AC)</i>	8
2.2.3 <i>Internet of Thing (IoT)</i>	8
2.2.4 Mikrokontroler.....	9
2.2.5 Mikrokontroler <i>Wemos D1 (R2)</i>	9
2.2.6 Sensor <i>Passive Infra Red (PIR)</i>	10
2.2.7 Sensor DHT22	11
2.2.8 <i>IR Transmitter</i>	12
2.2.9 <i>IR Receiver</i>	13
2.2.10 Sistem Monitoring	13
2.2.11 Protokol <i>Message Queue Telemetry Transport (MQTT)</i>	13

BAB III METODOLOGI PENELITIAN	15
3.1 Rencana Pelaksanaan	15
3.2 Analisis Kebutuhan Sistem	16
3.2.1 Analisis Kebutuhan Alat dan Bahan	17
3.2.2 Perencanaan Biaya	18
3.3 Perancangan Perangkat Keras	18
3.3.1 Perancangan Arsitektur Sistem	19
3.3.2 Rangkaian Elektronika Sistem	22
3.4 Perancangan Perangkat Lunak	24
3.4.1 Sistem Monitoring Berbasis <i>Web</i>	25
3.5 Rancangan Arsitektur MQTT Sistem	27
3.6 Implementasi Sistem	27
3.7 Pengujian dan Evaluasi Sistem	28
3.8 Dokumentasi dan Laporan	28
BAB IV HASIL DAN PEMBAHASAN	29
4.1 Realisasi Sistem	29
4.1.1 Realisasi Penyusunan Perangkat Keras	29
4.1.2 Realisasi Pembangunan Web	30
4.1.3 Realisasi <i>Interface</i> Sistem	31
4.1.4 Realisasi Pembangunan Program pada Mikrokontroler	33
4.1.5 Realisasi Pembangunan Arsitektur Komunikasi Data MQTT	40
4.2 Pengujian dan Evaluasi Sistem	42
4.2.1 Hasil Pengujian Perangkat Keras Sistem	42
4.2.2 Hasil Pengujian Sistem Web	45
4.2.3 Hasil Pengujian Fungsi Keseluruhan Sistem	46
4.2.4 Hasil Pengujian dengan <i>Metode Mean Opinion Score (MOS)</i>	48
BAB V PENUTUP	51
5.1 Kesimpulan	51
5.2 Saran	51
DAFTAR PUSTAKA	52

DAFTAR GAMBAR

Gambar 2. 1 Wemos D1 (R2).....	9
Gambar 2. 2 Sensor PIR	10
Gambar 2. 3 Diagram blok sensor PIR	11
Gambar 2. 4 Sensor DHT22	11
Gambar 2. 5 Sistem umum IoT memakai MQTT	14
Gambar 3. 1 Diagram alir tahap penelitian.....	15
Gambar 3. 2 Arsitektur sistem	19
Gambar 3. 3 Rangkaian elektronik sistem	21
Gambar 3. 4 Use <i>case</i> diagram sistem	23
Gambar 3. 5 Tampilan halaman <i>dashboard</i>	23
Gambar 3. 6 Tampilan halaman log Aktivitas	24
Gambar 3. 7 Tampilan halaman pengaturan	24
Gambar 3. 8 Rancangan arsitektur MQTT sistem	25
Gambar 4. 1 Realisasi Perangkat Keras	28
Gambar 4. 2 Folder Struktur pembangunan <i>web</i> dengan <i>CodeIgniter</i>	29
Gambar 4. 3 Halaman <i>dashboard</i>	30
Gambar 4. 4 Halaman log aktivitas	31
Gambar 4. 5 Halaman grafik penggunaan	31
Gambar 4. 6 Pengujian sensor DHT22	42
Gambar 4. 7 Lokasi penempatan perangkat keras sistem	45
Gambar 4. 8 Log aktivitas sistem	46

DAFTAR TABEL

Tabel 2. 1 Spesifikasi <i>Wemos D1 (R2)</i>	9
Tabel 3. 1 Perencanaan biaya	18
Tabel 3. 1 <i>IR code AC Sharp</i>	18
Tabel 3. 2 Perencanaan kondisi sistem	22
Tabel 4. 1 Hasil pengujian suhu udara sensor DHT22	42
Tabel 4. 2 Hasil Pengujian Kelembaban Udara Sensor DHT22	42
Tabel 4. 3 Hasil pengujian sensor PIR.....	43
Tabel 4. 4 Hasil pengujian <i>IR Transmitter</i>	43
Tabel 4. 5 Pengujian fungsi Log Aktivitas	44
Tabel 4. 6 Pengujian fungsi Grafik Penggunaan	45
Tabel 4. 7 Pengujian parameter MOS.....	47

DAFTAR LISTING

Listing 4.1 <i>Script Library control application</i>	32
Listing 4.2 <i>Script</i> pendeklarasian variabel <i>control</i> application	32
Listing 4.3 <i>Script</i> pendeklarasian variable <i>wifi</i>	34
Listing 4.4 <i>Script</i> setup jaringan <i>wifi</i>	34
Listing 4.5 <i>Script</i> fungsi <i>reconnect</i>	34
Listing 4.6 <i>Script</i> fungsi <i>setup</i>	35
Listing 4.7 <i>Script</i> fungsi <i>loop</i>	35
Listing 4.8 <i>Script</i> pembacaan sensor DHT22	36
Listing 4.9 <i>Script</i> penentuan kondisi	36
Listing 4.10 <i>Script</i> publish data	38
Listing 4.11 <i>Script</i> fungsi simpan data	38
Listing 4.12 <i>Script</i> fungsi simpan data <i>off</i>	38
Listing 4.13 <i>Script</i> fungsi koneksi broker.....	39
Listing 4.14 <i>Script</i> fungsi <i>lost connection</i>	39
Listing 4.15 <i>Script</i> fungsi penerimaan pesan.....	39
Listing 4.16 <i>Script</i> deklarasi variabel <i>options</i>	40

ABSTRAK

Salah satu faktor kenyamanan adalah suhu dan kelembaban udara. Menurut SNI Termal rentang suhu nyaman berada pada 20.5°C-27.1°C. Sehingga apabila suhu berada di atas 27.1°C maka diperlukan *Air Conditioner* (AC) untuk mendapatkan kenyamanan termal yang efektif. Pengoperasian AC yang terpasang pada setiap ruang kelas PSTI FT UNRAM dilakukan secara manual, yang menyebabkan penggunaan AC yang tidak efisien. Penelitian ini menerapkan *Internet of Things* (IoT) untuk manajemen AC secara otomatis. 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 sehingga ruang kelas dapat disebut sebagai *adaptive classroom*. Dari pengujian MOS yang telah dilakukan diperoleh nilai sebesar 4.57 dari skala 5 yang menunjukkan bahwa sistem telah layak digunakan dan termasuk dalam kategori sangat baik.

Kata kunci: *Air Conditioner, Internet of Things, DHT22, PIR, MQTT.*

ABSTRACT

Temperature and humidity are one of the comfort factors. According to the SNI standard, the comfortable temperature range is at 20.5°C-27.1°C. If the temperature is above 27.1°C, an Air Conditioner (AC) is needed to get the effective thermal comfort. The air conditioners installed in each PSTI FT UNRAM classroom is operated manually. This research implements Internet of Things (IoT) for automatic AC usage management. This IoT device is made using a PIR sensor to detect human movements and DHT22 sensor to detect temperature and humidity of the air, as well as the MQTT protocol for data communication between the IoT device and the web system. Based on the testing scenario for the classroom containing students and the empty classroom that has been done, the IoT device is able to manage the AC usage automatically, so that the classroom can be called an adaptive classroom. From the MOS testing that has been done, a value of 4.57 from a scale of 5 is obtained, which shows that the system is feasible to use and is included in very good category.

Key word: Air Conditioner, Internet of Things, DHT22, PIR, MQTT

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kenyamanan merupakan hal yang sangat penting untuk diperhatikan terkhusus pada saat melakukan aktivitas kegiatan belajar mengajar di dalam ruang kelas. Salah satu faktor kenyamanan adalah suhu dan kelembaban udara. Menurut SNI Termal rentang suhu Sejuk-Nyaman berada pada suhu 20.5°C-22.8°C dengan kelembaban relatif 50%-80%, kemudian rentang suhu Nyaman-Optimal 22.8°C-25.8°C dengan kelembaban relatif 70%-80%, dan rentang suhu Hampir Nyaman 25.8°C-27.1°C dengan kelembaban relatif 60%-70%. Sehingga apabila suhu berada di atas 27.1°C maka diperlukan alat pendingin udara seperti *Air Conditioner* (AC) untuk mendapatkan kenyamanan termal yang efektif dan dapat memberikan pengaruh positif dalam kinerja belajar mengajar di dalam ruangan [1].

Pengoperasian AC yang terpasang pada setiap ruang kelas PSTI FT UNRAM (Program Studi Teknik Informatika Fakultas Teknik Universitas Mataram) masih dilakukan secara manual, yaitu menggunakan *remote control* yang terdapat pada setiap ruang kelas. Ketika kegiatan perkuliahan telah selesai, sering kali dijumpai AC dalam keadaan menyala. Hal ini disebabkan karena kelalaian pengguna ruangan. Kejadian ini akan membuat pemborosan energi listrik jika terus menerus dilakukan. Oleh karena itu, perlu adanya sistem kontrol suhu ruangan pada ruang kelas PSTI FT UNRAM yang lebih praktis dan efisien.

Lingkungan PSTI FT UNRAM telah tersedia jaringan wifi yang mampu menjangkau semua ruangan kelas yang ada pada PSTI FT UNRAM. Jaringan wifi ini dapat digunakan untuk membuat sistem kontrol otomatis AC dengan memanfaatkan konsep *Internet of Things* (IoT). Teknologi IoT merupakan bentuk koneksi suatu perangkat yang saling terhubung dan mampu menghasilkan suatu informasi yang dapat diakses dan digunakan oleh manusia atau sistem lainnya [2]. Perkembangan teknologi IoT memberikan banyak sekali kemudahan seperti halnya dalam melakukan kendali peralatan rumah dari jarak jauh, mengontrol dan memantau secara *real time* dengan menggunakan *smartphone* atau melalui desktop [3f].

Message Queue Telemetry Transport (MQTT) adalah sebuah protokol yang dapat digunakan untuk menerapkan konsep IoT. MQTT dirasa tepat untuk menjadi protokol IoT karena MQTT bersifat *light weighted message* dan didesain untuk perangkat yang

memiliki sumber daya terbatas [4]. Protokol MQTT merupakan sebuah protokol komunikasi data *machine to machine* (M2M) yang bersifat ringan, terbuka, sederhana dan didesain agar mudah diimplementasikan. Protokol MQTT memiliki keunggulan yaitu dapat mengirim data dengan *bandwidth* yang rendah serta menjamin semua pesan terkirim meski koneksi terputus sementara [5].

Berdasarkan hal-hal tersebut pada penelitian ini akan dibangun sistem manajemen penggunaan AC secara otomatis dengan memanfaatkan konsep IoT. Sistem tersebut akan diatur secara otomatis dengan sensor DHT22 yang akan membaca suhu dan kelembaban dalam ruangan, sensor PIR sebagai sensor pendeteksi keberadaan seseorang dan IR *Transmitter* sebagai transmisi instruksi otomatis yang dikirimkan oleh mikrokontroler ke AC. Aktivitas sistem akan dipantau menggunakan sistem berbasis *web* dengan protokol komunikasi MQTT.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan didapat rumusan masalah antara lain sebagai berikut :

1. Bagaimana merancang rangkaian elektronik perangkat sistem manajemen penggunaan AC secara otomatis berbasis IoT pada ruang A3-02 PSTI FT UNRAM?
2. Bagaimana membangun sistem berbasis *web* sebagai media monitoring sistem manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM?
3. Bagaimana mengimplementasikan protokol MQTT sebagai protokol komunikasi data pada sistem manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM?

1.3. Batasan Masalah

Adapun batasan masalah yakni sebagai berikut :

1. Implementasi dan pengujian sistem dilakukan pada ruang A3-02 PSTI FT UNRAM.
2. Otomatisasi yang dilakukan yaitu dapat menghidupkan, mengontrol suhu dan mematikan AC sesuai dengan kondisi suhu udara yang dideteksi.
3. Interval suhu nyaman disesuaikan dengan SNI Termal 14-1993-03.
4. Tidak menghitung jumlah daya yang digunakan oleh AC.

1.4. Tujuan

Adapun tujuan yang dilakukan pada penelitian tugas akhir ini sebagai berikut :

1. Merancang rangkaian elektronik perangkat sistem manajemen penggunaan AC secara otomatis berbasis IoT pada ruang A3-02 PSTI FT UNRAM.
2. Membangun sistem berbasis *web* sebagai media *monitoring* sistem manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM.
3. Mengimplementasikan protokol MQTT sebagai protokol komunikasi data sistem manajemen penggunaan AC secara otomatis berbasis IoT pada ruang A3-02 PSTI FT UNRAM.

1.5. Manfaat

Manfaat dari penelitian tugas akhir ini sebagai berikut :

1. Mempermudah manajemen penggunaan AC pada ruang A3-02 PSTI FT UNRAM.
2. Menghindari *human error* dalam manajemen penggunaan AC yang mengakibatkan penggunaan tidak efisien.
3. Dapat mencatat penggunaan AC secara periodik.
4. Dapat memantau penggunaan AC secara *real time* dengan sistem berbasis *web*.

1.6. Sistematika Penulisan

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

- Bab I. Pendahuluan

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

- 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.

- 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.

- Bab IV. Hasil dan Pembahasan

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

- 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 tinjauan pustaka dari penelitian yang terkait yang telah dilakukan sebelumnya, sebagai bahan acuan untuk melakukan penelitian ini

Pada penelitian dengan judul Rancang Bangun Sistem *Control Air Conditioning* Automatis Berbasis *Passive Infrared Receiver*, peneliti merancang suatu sistem perangkat keras yang *portable* yaitu sistem kontrol *Air Conditioning* (AC) secara otomatis menggunakan *microcontroller ATmega16* dengan sensor *Passive Infrared* (PIR) sebagai sensor pendeteksi gerak dan sensor LM32 sebagai sensor pendeteksi suhu. Dalam perancangan sistem ini, telah berhasil dibuat sistem kontrol model atau *prototipe* kontrol AC dan telah berfungsi dengan baik. Berdasarkan pengujian yang telah dilakukan, sistem kontrol AC otomatis, sudah dapat melakukan ON/OFF otomatis pada AC dengan baik dan akurat [6].

Pada penelitian sistem kontrol AC secara otomatis berbasis sensor PIR dengan penelitian *Adaptive Classroom* Berbasis IoT Manajemen Penggunaan AC Secara Otomatis memiliki persamaan pada penggunaan sensor PIR dan bertujuan untuk melakukan otomatisasi pada alat elektronik AC. Sedangkan untuk perbedaannya sendiri pada penelitian ini akan menggunakan sensor DHT22 dan mikrokontroler *Wemos D1 (R2)*. Sistem akan memanfaatkan konsep IoT dengan protokol MQTT dan dilakukan pencatatan data ke dalam *database* yang dapat dimonitor menggunakan sistem *web*.

Pada penelitian lainnya yang berjudul Sistem Kendali Suhu dan Pemantauan Kelembaban Udara Ruangan Berbasis *Arduino Uno* dengan Menggunakan Sensor DHT22 dan *Passive Infrared* (PIR), peneliti membuat sistem kendali suhu dan pemantauan kelembaban udara ruangan berbasis *Arduino Uno* dengan menggunakan sensor DHT22 dan sensor PIR. Perangkat ini terhubung dengan sebuah kipas angin yang dapat menstabilkan suhu ruangan ketika suhu tersebut dinilai kurang nyaman untuk melakukan berbagai kegiatan di dalam ruangan. Setelah dilakukan perancangan alat dan pengujian terhadap komponen-komponen yang digunakan, dapat disimpulkan bahwa alat telah berhasil dibuat dan sesuai dengan tujuan pembuatan. *Relay* dalam kondisi aktif (kipas menyala) ketika suhu di atas 28°C. Proses *monitoring* suhu dilakukan secara lokal yaitu dengan menggunakan modul LCD dan RTC sebagai tampilan suhu dan kelembaban secara *real time* [7].

Persamaan penelitian tentang Sistem Kendali Suhu dan Pemantauan Kelembaban Udara Ruang Berbasis *Arduino Uno* dengan Menggunakan Sensor DHT22 dan PIR dengan penelitian *Adaptive Classroom* Berbasis IoT Manajemen Penggunaan AC Secara Otomatis yaitu pada penggunaan sensor gerak PIR dan sensor DHT22. Perbedaannya sendiri pada penelitian ini menggunakan AC sebagai penstabil suhu dalam ruangan dengan perintah kontrol yang dikirim menggunakan IR *Transmitter* dengan mikrokontroler *Wemos D1 (R2)*. Sistem akan memanfaatkan konsep IoT dengan protokol komunikasi MQTT dan akan melakukan pencatatan data ke dalam *database* yang dapat di *monitoring* menggunakan sistem *web*.

Pada penelitian yang berjudul Pengendalian *Air Conditioner* Jarak Jauh Menggunakan *Arduino* dan *WiFi*, peneliti membuat sistem pada ruang kuliah Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta yaitu sistem pengendalian AC dari jarak jauh menggunakan mikrokontroler *Arduino Mega 2560* dan jaringan *WiFi* sebagai jalur komunikasi. Sensor yang digunakan yaitu sensor suhu *LM35* dan sensor *LDR* untuk mendeteksi intensitas cahaya dari lampu indikator AC, apakah AC dalam keadaan status ON atau status OFF. Untuk menerima dan merekam sinyal IR *code* dari *remote* kontrol AC digunakan IR *Receiver*, sedangkan IR *Transmitter* digunakan untuk mengirim sinyal IR *code* ke AC. Sistem akan dikendalikan menggunakan *web* yang dapat diakses menggunakan *smartphone* atau komputer [8].

Pada penelitian Pengendalian *Air Conditioner* Jarak Jauh Menggunakan *Arduino* dan *WiFi* dengan penelitian *Adaptive Classroom* Berbasis IoT, Manajemen Penggunaan AC Secara Otomatis memiliki persamaan dalam melakukan pengambilan IR *code* yaitu menggunakan IR *Receiver* dan IR *Transmitter* sebagai transmisi sinyal IR *code* ke AC. Selain hal tersebut untuk melakukan pengoperasian atau *monitoring* menggunakan sistem *web*. Perbedaannya sendiri pada penelitian ini akan melakukan otomatisasi manajemen penggunaan AC dengan menggunakan sensor PIR dan sensor DHT22. Sistem akan dipantau melalui *web* menggunakan protokol komunikasi MQTT dengan mikrokontroler *Wemos D1 (R2)*.

Pada penelitian yang berjudul Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada *Smartphone*, peneliti membuat sistem kendali berbasis mikrokontroler *Wemos D1 (R2)* dengan protokol komunikasi MQTT pada *smart home*. Peneliti membuat contoh sederhana dari *smart home*, dimana 2 buah lampu LED akan mati/menyala berdasarkan dari nilai data yang didapatkan dari sensor. Sensor yang

digunakan adalah sensor suhu DHT11 dan sensor LDR. Sistem dikontrol melalui aplikasi sistem kendali. Selain itu aplikasi dapat menampilkan data sensor suhu dan cahaya dalam bentuk grafik sebagai fungsi *monitoring* dari sistem kendali. Berdasarkan hasil pengujian keandalan sistem nilai integritas data yang dikirim dan diterima melalui protokol MQTT adalah 100% [4].

Pada penelitian Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada *Smartphone* dengan penelitian *Adaptive Classroom* Berbasis IoT Manajemen Penggunaan AC Secara Otomatis memiliki persamaan pada penggunaan mikrokontroler Wemos D1 (R2) dengan protokol MQTT dan sistem akan menampilkan data dalam bentuk grafik sebagai fungsi *monitoring* sistem. Sedangkan untuk perbedaannya sendiri pada penelitian ini akan menggunakan sensor PIR dan DHT22 untuk melakukan otomatisasi perangkat elektronik AC. Sistem akan dipantau menggunakan sistem *web*.

2.2. Dasar Teori

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

2.2.1 Nyaman Termal

Kenyamanan adalah interaksi dan reaksi manusia terhadap lingkungan yang bebas dari rasa negatif dan bersifat subjektif. Kenyamanan termal adalah sebuah kondisi pemikiran yang mengekspresikan kepuasan atas lingkungan termal nya. Faktor-faktor kenyamanan termal di dalam ruang tertutup adalah temperatur udara, kelembaban udara, temperatur radiasi rata – rata dari atap dan dinding, kecepatan gerak udara, tingkat pencahayaan dan distribusi cahaya pada dinding pandangan. Selain itu terdapat beberapa standar yang menentukan kenyamanan termal. Standar Nasional Indonesia Termal SNI T-14-1993-03 yaitu rentang suhu sejuk-nyaman 20.5°C-22.8°C dengan kelembaban relatif 50%-80%, rentang suhu nyaman-optimal 22.8°C-25.8°C dengan kelembaban relatif 70%-80%, dan rentang suhu hampir nyaman 25.8°C-27.1°C dengan kelembaban relatif 60%-70% [1].

Temperatur udara merupakan salah satu faktor yang paling dominan dalam menentukan kenyamanan termal. Manusia dikatakan nyaman apabila suhu tubuhnya sekitar 37° *Celcius*. Temperatur tubuh manusia sangat dipengaruhi oleh temperatur yang berada di sekitarnya, dalam hal ini akan berlaku hukum termodinamika yang menyatakan

bahwa temperatur akan berpindah dari tempat dengan temperatur tinggi ke tempat dengan temperatur rendah. Selain temperatur udara, kelembaban udara juga memiliki pengaruh yang besar dalam menentukan kenyamanan termal. Kelembaban di sebuah tempat akan mempengaruhi terjadinya perpindahan panas dari dan menuju tubuh. Lingkungan yang mempunyai kelembaban relatif tinggi mencegah penguapan keringat dari kulit di lingkungan yang panas, semakin sedikit keringat yang menguap karena kelembaban tinggi, sehingga kegerahan bagi individu yang berada di lingkungan tersebut [9].

2.2.2 Air Conditioner (AC)

Suatu sistem pengkondisi udara, atau yang sering disebut HVAC (*Heating, Ventilating, Air Conditioning*) meliputi suatu sistem motor, saluran pipa, sistem pertukaran udara, sistem kontrol dan sistem pertukaran panas. Tujuan sistem HVAC ini adalah untuk mengatur aliran udara serta memindahkan komponen udara yang tidak diinginkan dari suatu ruangan atau fasilitas dengan tujuan untuk mendapatkan kondisi yang diinginkan oleh pengguna atau peralatan yang ada. Menyediakan mutu udara yang baik dalam ruangan adalah suatu fungsi utama dari sistem HVAC, mengatur pergerakan udara untuk mengusir bau dan debu agar kenyamanan dan kesehatan tetap terjaga [6].

2.2.3 Internet of Thing (IoT)

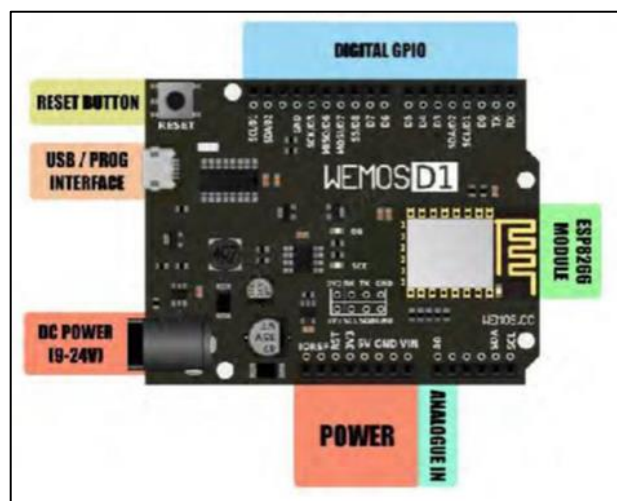
Internet of Things (IoT) merupakan suatu jaringan yang menghubungkan berbagai objek yang memiliki identitas pengenalan serta alamat IP, sehingga dapat saling berkomunikasi dan bertukar informasi mengenai dirinya maupun lingkungan yang diinderanya. Objek dalam IoT dapat menggunakan maupun menghasilkan layanan-layanan dan saling bekerjasama untuk mencapai suatu tujuan bersama. Dengan kemampuannya ini, IoT telah menggeser definisi internet sebagai komputasi dimana saja kapan saja bagaimana saja, menjadi apa saja siapa saja dan layanan apa saja [10]. Tujuan *internet of things* adalah untuk membuat manusia berinteraksi dengan benda lebih mudah, bahkan dengan tujuan supaya benda juga bisa saling berkomunikasi antar satu benda dengan benda yang lainnya. *Internet of things* pada dasarnya adalah memulai sistem dengan cara membuat suatu koneksi atau sebuah komunikasi antara mesin dengan mesin, sehingga mesin – mesin tersebut dapat berinteraksi dan dapat bekerja secara independen sesuai dengan data yang diperoleh, yang kemudian dapat mengolahnya secara mandiri. Sistem *internet of thing* dapat tumbuh menjadi lebih kompleks dengan sejumlah besar perangkat yang dapat saling berhubungan untuk menghasilkan layanan yang mendukung untuk proses yang lebih kompleks [3].

2.2.4 Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer fungsional yang di letakkan dalam papan elektronik yang berukuran *mikro* atau kecil. Di dalam mikrokontroler terdapat sebuah prosesor, memori, serta komponen *input* dan *output*. Dengan begitu, mikrokontroler adalah suatu alat elektronika digital yang memiliki *input* serta *output* yang kendalinya dapat diprogram ulang dengan suatu cara khusus. Ada banyak pilihan mikrokontroler yang telah terdapat modul jaringan agar dapat langsung terhubung ke jaringan internet sehingga dapat diimplementasikan untuk membuat peralatan berbasis IoT [11].

2.2.5 Mikrokontroler Wemos D1 (R2)

Mikrokontroler Wemos D1 (R2) berbasis ESP8266 yaitu sebuah modul mikrokontroler nirkabel WiFi 802.11 yang kompatibel dengan Arduino IDE. Tata letak mikrokontroler ini didasarkan pada desain *hardware Arduino* standar dengan proporsi yang sama dengan *Arduino Uno* dan *Leonardo*. Mikrokontroler ini juga sudah termasuk satu set *header Arduino* standar yang artinya kompatibel dengan beragam *Arduino shield* [12]. Bentuk fisik dari Wemos D1 (R2) sendiri dapat dilihat pada Gambar 2.1.



Gambar 2.1 Wemos D1 (R2)

Tabel 2.1 Spesifikasi Wemos D1 (R2)

Perangkat Keras	Spesifikasi
Mikrokontroler	ESP-8266EX
<i>Operating Voltage</i>	3.3 V
Digital I/O Pins	11
Analog Input Pin	1
Clock Speed	80MHz/160MHz

<i>Flash Memori</i>	4 MB
<i>Length</i>	68.6mm
<i>Width</i>	53.4mm
<i>Weight</i>	25g

2.2.6 Sensor *Passive Infra Red* (PIR)

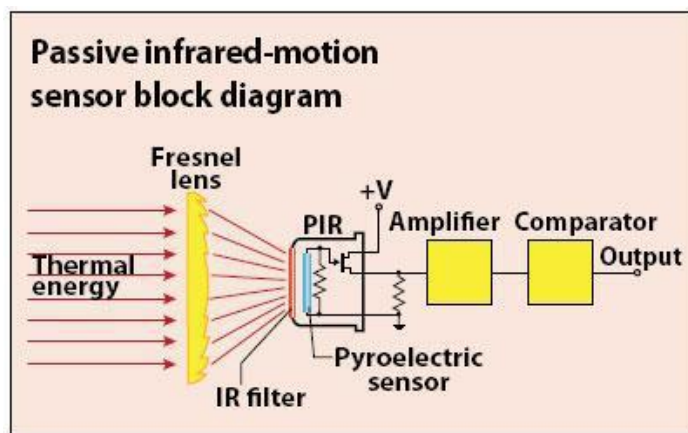
Sensor PIR (*Passive Infra Red*) adalah sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah. Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi menerima radiasi sinar infra merah dari luar. Sensor PIR ini digunakan dalam perancangan detektor gerak. Karena semua benda memancarkan energi radiasi, sebuah gerakan akan terdeteksi ketika sumber infra merah dengan suhu tertentu seperti manusia melewati sumber infra merah yang lain dengan suhu yang berbeda seperti dinding, maka sensor akan membandingkan pancaran infra merah yang diterima setiap waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor [13]. Bentuk fisik dari sensor PIR (*Passive Infra Red*) sendiri dapat dilihat pada Gambar 2.2.



Gambar 2.2 Sensor PIR

Cara kerja dari sensor PIR ini yaitu ketika pancaran infra merah masuk melalui lensa *Fresnel* dan mengenai sensor *pyroelectric*, karena sinar infra merah mengandung energi panas maka sensor *pyroelectric* akan menghasilkan arus listrik. IR Filter yang ada pada sensor PIR mampu menyaring panjang gelombang sinar inframerah pasif antara 8 – 14 mikrometer. Sedangkan jarak jangkauan maksimum dari sensor PIR ini adalah ± 7 meter dan sensor ini sangat efektif digunakan sebagai *human detector* karena manusia sendiri memiliki suhu badan yang dapat menghasilkan pancaran infra merah dengan panjang gelombang antara 9-10 mikrometer. Ketika seseorang berjalan melewati sensor, sensor akan menangkap pancaran sinar inframerah pasif yang dipancarkan oleh tubuh manusia

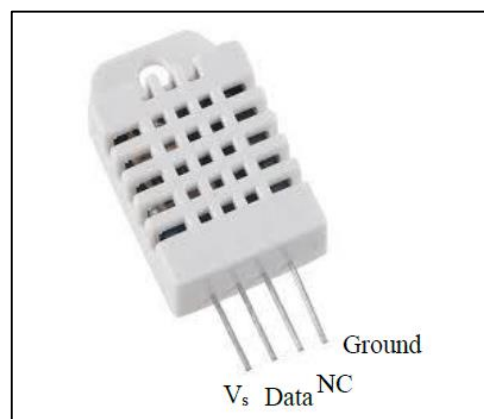
yang memiliki suhu yang berbeda dari lingkungan sehingga menyebabkan material *pyroelectric* bereaksi dan menghasilkan arus listrik. Kemudian sebuah sirkuit *amplifier* yang ada menguatkan arus tersebut yang kemudian dibandingkan oleh *comparator* sehingga menghasilkan output. Sedangkan ketika seseorang berada di depan sensor PIR dengan kondisi diam, maka sensor PIR akan menghitung panjang gelombang yang dihasilkan oleh tubuh manusia tersebut. Panjang gelombang yang konstan ini menyebabkan energi panas yang dihasilkan dapat digambarkan hampir sama pada kondisi lingkungan sekitarnya dan sensor tidak akan merespons apapun [14]. Untuk lebih jelasnya dapat dilihat diagram blok sensor PIR pada Gambar 2.3.



Gambar 2.3 Diagram blok sensor PIR

2.2.7 Sensor DHT22

DHT-22 merupakan salah satu sensor suhu dan kelembaban yang juga dikenal sebagai sensor AM2302. Sensor ini memiliki empat kaki yaitu kaki V_s , Data, NC dan *Ground*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.4.



Gambar 2.4 Sensor DHT22

Pada Gambar 2.4 tegangan sumber disambungkan ke kaki V_s dimana tegangan sumber yang digunakan pada umumnya adalah sebesar 5V karena mengikuti tegangan

kerja mikrokontroler yaitu sebesar 5V. Kemudian kaki Data disambungkan dengan sebuah mikrokontroler yang digunakan untuk mengambil data suhu dan kelembaban udara yang telah diukur. Kaki NC yaitu kaki *Not Connected*, merupakan kaki yang tidak disambungkan kemampuan. Sedangkan kaki *Ground* disambung dengan *Ground* tegangan sumber [15]. Spesifikasi teknis DHT22 secara keseluruhan dapat dijabarkan sebagai berikut:

1. Rentang catu daya: 3,3 – 6 Volt DC (tipikal 5 VDC)
2. Konsumsi arus pada saat pengukuran antara 1 hingga 1,5 mA
3. Sinyal keluaran: digital lewat bus tunggal dengan kecepatan 5 MS / operasi (*MSB-first*)
4. Elemen pendeteksi: kapasitor polimer (*polymer capacitor*)
5. Jenis sensor: kapasitif (*capacitive sensing*)
6. Rentang deteksi kelembaban: 0-100% RH (akurasi $\pm 2\%$ RH)
7. Rentang deteksi suhu: $-40^{\circ} \sim +80^{\circ}$ Celcius (akurasi $\pm 0,5^{\circ}\text{C}$)
8. Resolusi sensitivitas: 0,1% RH; $0,1^{\circ}\text{C}$
9. Pengulangan: $\pm 1\%$ RH; $\pm 0,2^{\circ}\text{C}$
10. *Hysteresis* kelembaban: $\pm 0,3\%$ RH
11. Stabilitas jangka panjang: $\pm 0,5\%$ RH / tahun
12. Periode pemindai rata-rata: 2 detik
13. Ukuran: 25,1 x 15,1 x 7,7 mm

2.2.8 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 sistem kendali jarak jauh, sistem pengamanan, dan otomatisasi pada sistem. 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 *remote AC*. Cahaya LED timbul sebagai akibat penggabungan *electron* dan *hole* pada persambungan antara dua jenis *semikonduktor* dimana setiap penggabungan disertai dengan pelepasan energy. Pada penggunaan LED *infrared* dapat diaktifkan dengan tegangan *Direct Current* (DC) untuk transmisi atau sensor jarak dekat, dan dengan tegangan *Alternating Current* (AC) (30-40 kHz) untuk transmisi atau sensor jarak jauh [16].

2.2.9 IR Receiver

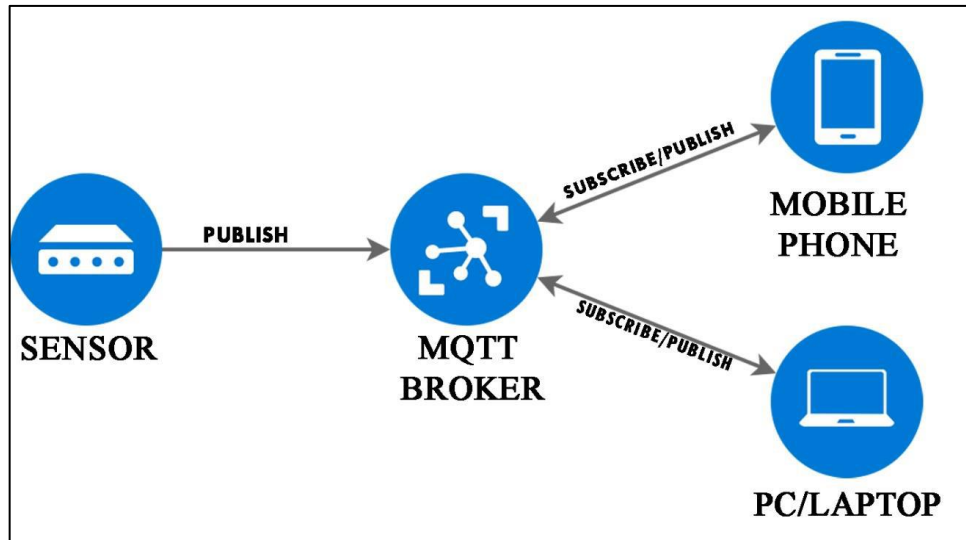
Infrared receiver merupakan suatu modul penerima data melalui gelombang infra merah dengan *frekuensi carrier* sebesar 38 kHz. Modul ini dapat difungsikan sebagai input dalam aplikasi transmisi data nirkabel seperti sistem kendali jarak jauh, sistem pengamanan, dan otomatisasi pada sistem. *Receiver* (penerima) yang digunakan untuk sensor infra merah adalah jenis *photo transistor*, yaitu jenis *transistor* bipolar yang menggunakan kontak (*junction*) *base-collector* untuk menerima atau mendeteksi 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 [16].

2.2.10 Sistem Monitoring

Sistem *monitoring* merupakan sebuah sistem yang digunakan untuk melakukan pemantauan, pengamatan, dan pengawasan. *Monitoring* bertujuan untuk memberikan suatu informasi keberlangsungan proses dalam 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 [17].

2.2.11 Protokol Message Queue Telemetry Transport (MQTT)

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 [4]. *Publish* atau *subscribe* sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana pengirim data disebut *publisher* dan penerima data disebut dengan *subscriber*. Selain itu juga terdapat sebuah entitas yang berfungsi untuk menangani *publish* dan *subscribe* data yaitu *Broker* [18]. Untuk lebih jelasnya dapat dilihat pada Gambar 2.5.



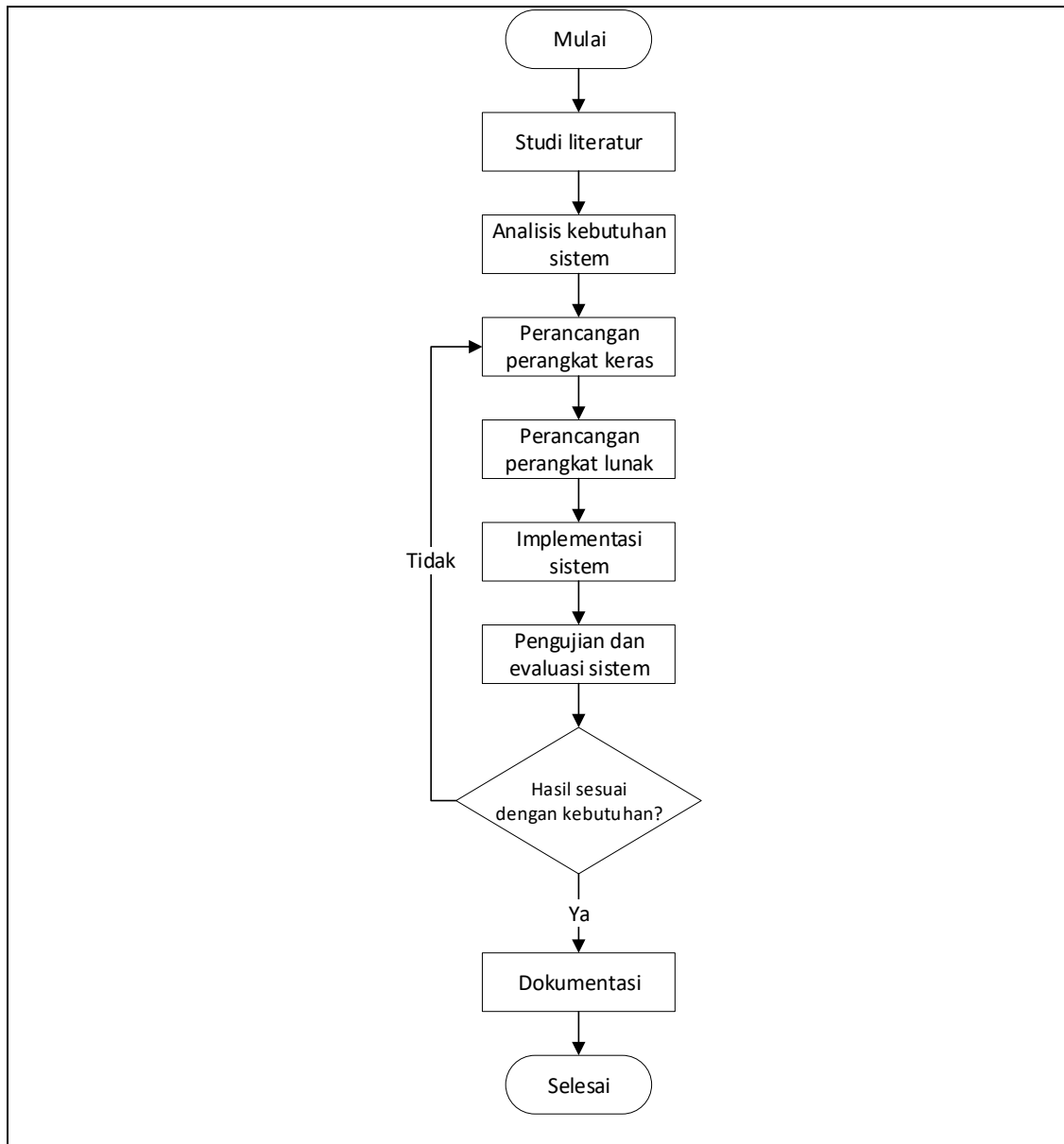
Gambar 2.5 Sistem umum IoT memakai MQTT

BAB III

METODOLOGI PENELITIAN

3.1 Rencana Pelaksanaan

Adapun rencana pelaksanaan penelitian sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir tahap penelitian

Pada Gambar 3.1 merupakan alur dari rencana pelaksanaan penelitian sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis. Penjelasan untuk masing-masing proses pada Gambar 3.1 adalah sebagai berikut:

1. Pada tahap studi literatur, dilakukan studi dan pengkajian terhadap penelitian-penelitian yang sudah dilaksanakan sebelumnya dan penelitian lain yang terkait

dengan penelitian yang akan dilaksanakan. Pengkajian ini dilakukan untuk mengetahui hasil dari penelitian yang telah dilakukan sebelumnya, sehingga dapat dijadikan bahan acuan atau referensi untuk penelitian yang akan dilaksanakan.

2. Pada tahap analisis kebutuhan sistem, dilakukan analisis terhadap kebutuhan dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang akan dibangun, seperti analisis kebutuhan masukan dan keluaran dari sistem serta analisis perangkat apa saja yang dibutuhkan dalam proses perancangan dan pembangunan sistem.
3. Pada tahap perancangan perangkat keras, dilakukan perancangan terhadap keseluruhan rangkaian perangkat keras sistem berupa arsitektur dan rangkaian elektronika dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang akan dibangun.
4. Pada tahap perancangan perangkat lunak, akan berkaitan dengan perancangan protokol MQTT dan sistem berbasis *web* sebagai media *monitoring* suhu, kelembaban dan kondisi sistem.
5. Pada tahap implementasi, dilakukan penyusunan perangkat dan penerapan protokol MQTT serta pembuatan sistem berbasis *web* dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis.
6. Pada tahap pengujian dan evaluasi sistem, dilakukan pengujian terhadap sistem yang sudah dibangun dan melakukan evaluasi dari hasil pengujian yang dilakukan tersebut. Jika sistem sudah 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.
7. Pada tahap dokumentasi, akan dibuat laporan dari hasil pengujian dan evaluasi sistem yang telah dilakukan sebelumnya.

3.2 Analisis Kebutuhan Sistem

Pada tahap analisis kebutuhan sistem akan dilakukan analisis terhadap kebutuhan dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis. Analisis yang dilakukan meliputi analisis kebutuhan alat dan bahan yang diperlukan dan anggaran biaya dalam pembuatan sistem.

3.2.1 Analisis Kebutuhan Alat dan Bahan

Dalam perancangan sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis, ada beberapa alat dan bahan yang dibutuhkan, yaitu:

1. Alat Penelitian

Alat – alat yang diperlukan dalam penelitian tugas akhir ini dibagi menjadi dua yakni perangkat keras dan perangkat lunak antara lain sebagai berikut:

a. Perangkat Keras

- Laptop/PC untuk aktivitas pembuatan *code* aplikasi
- AC Sharp yang sudah terpasang pada ruang A3-02 PSTI FT UNRAM
- *Remote* AC Sharp untuk mengambil IR *code* yang digunakan untuk mengontrol AC

b. Perangkat Lunak

- Sistem operasi Windows 10
- Arduino IDE versi 1.8.9 sebagai *code* editor untuk mikrokontroler
- *Sublime Text* versi 3.2.1 sebagai *code* editor untuk *web*
- *Browser* untuk mengakses halaman *web*

2. Bahan Penelitian

Adapun bahan – bahan yang diperlukan dalam penelitian tugas akhir ini sebagai berikut:

- 1 buah *Wemos D1 (R2)* sebagai mikrokontroler.
- 1 buah sensor DHT22 sebagai pendeteksi suhu dan kelembaban udara.
- 1 buah sensor PIR sebagai pendeteksi keberadaan seseorang.
- 2 buah IR *Transmitter* sebagai transmisi sinyal *code* ke AC.
- 1 buah IR *Receiver* untuk merekam IR *code* dari *remote* AC
- 2 buah AC Sharp yang sudah terpasang pada ruangan A3-02 PSTI FT UNRAM.
- 1 buah *breadboard*
- 1 set kabel *jumper*
- Remote* AC

3.2.2 Perencanaan Biaya

Anggaran biaya pada Tabel 3.1 merupakan anggaran biaya yang akan digunakan untuk membeli alat dan bahan sesuai dengan kebutuhan pada analisis kebutuhan alat dan bahan.

Tabel 3.1 Perencanaan biaya

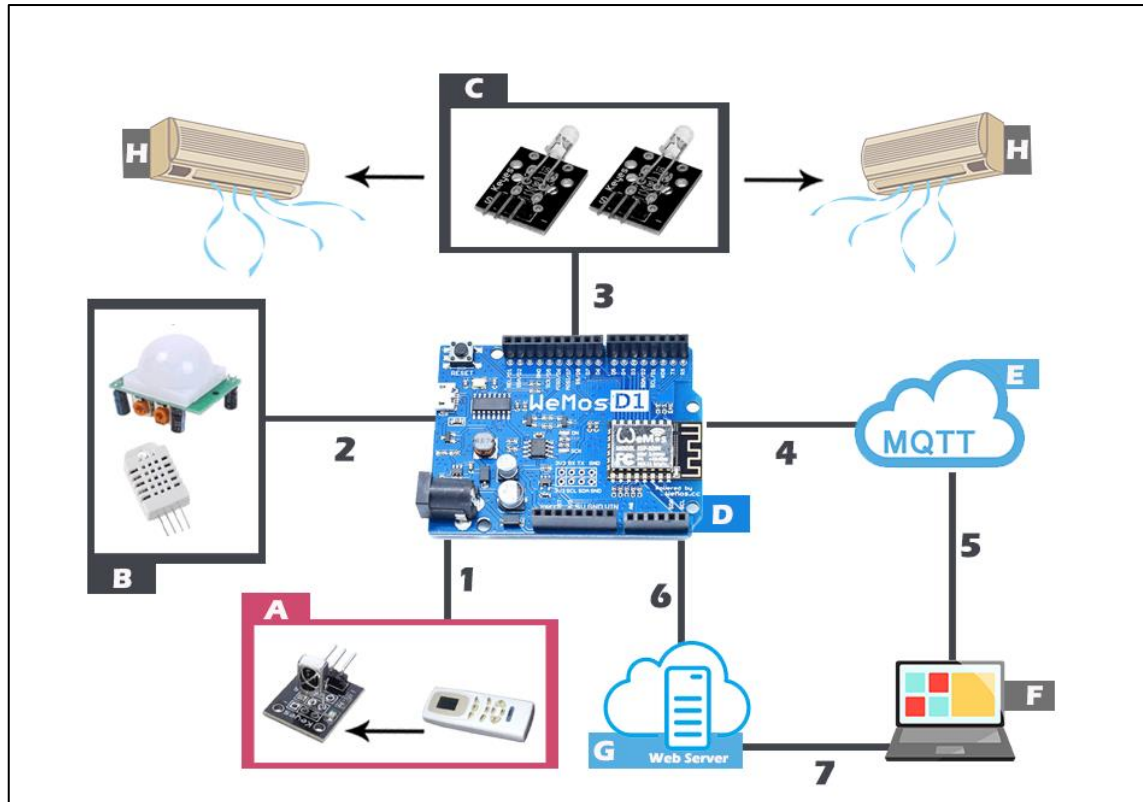
No	Nama Alat	Jumlah	Harga
1.	<i>Wemos D1 (R2)</i>	1 buah	Rp. 100.000
2.	DHT22	1 buah	Rp. 60.000
3.	Sensor PIR	1 buah	Rp. 20.000
4.	Kabel <i>Jumper</i>	1 set	Rp. 20.000
5.	<i>Breadboard</i>	1 buah	Rp. 30.000
6.	IR <i>Transmitter</i> KY-005	2 buah	Rp. 30.000
7.	IR <i>Receiver</i> KY-022	1 buah	Rp. 15.000
Jumlah			RP. 275.000

3.3 Perancangan Perangkat Keras

Pada tahap perancangan perangkat keras, akan dilakukan perancangan terhadap arsitektur sistem dan gambaran rangkaian dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis.

3.3.1 Perancangan Arsitektur Sistem

Gambaran dari arsitektur sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang akan dibangun dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur sistem

Gambar 3.2 merupakan gambaran arsitektur dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang akan dibuat pada penelitian ini. Berikut merupakan penjelasan dari masing-masing proses yang terdapat pada Gambar 3.2.

1. IR Receiver (A) pada rangkaian elektronik digunakan untuk mengambil IR code dari remote AC.
2. Sensor (B) adalah sensor PIR yang digunakan untuk mendeteksi pancaran sinar inframerah yang dihasilkan oleh gerakan manusia dan sensor DHT22 yang digunakan untuk mendeteksi suhu dan kelembaban udara.
3. IR Transmitter I pada rangkaian elektronik digunakan untuk mengirim sinyal perintah ke AC berdasarkan instruksi yang telah diolah oleh mikrokontroler (D).
4. Mikrokontroler Wemos D1 (R2) (D) digunakan untuk mengambil data dari sensor, 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 ESP8266HTTPClient* yang digunakan untuk dapat mengakses layanan *web*.

5. *Server Broker I* merupakan *server* yang akan dituju oleh *client* dalam penyampaian data dimana pengiriman datanya menggunakan *protocol* MQTT.
6. Laptop/PC (F) perangkat yang digunakan untuk mengakses sistem *monitoring* berbasis *web*.
7. *Web Server* (G) digunakan untuk menyimpan halaman *web* sekaligus sebagai tempat penyimpanan hasil pencatatan data
8. *Air Conditioner* (H) perangkat elektronik yang akan dikendalikan oleh sistem.

Berikut merupakan penjelasan dari hubungan antar perangkat yang terdapat pada Gambar 3.2.

1. Pengambilan *IR code* dari *remote* AC menggunakan *IR Receiver* diterima oleh mikrokontroler (D) yang akan dijadikan sebagai *code* perintah untuk mengontrol AC.
2. Hasil dari pembacaan sensor (B) diterima oleh mikrokontroler (D) untuk melakukan pengolahan dalam menentukan kondisi yang akan dikirimkan ke AC (H) menggunakan *IR Trasmitter* (C).
3. Mikrokontroler (D) mengirim nilai sensor (B) dan status AC yang didapat menuju *server broker I*.
4. Mikrokontroler (D) mengirim nilai sensor (B) dan status AC yang didapat menuju *server broker I*.
5. Mikrokontroler juga mengirim data hasil penentuan kondisi dengan mengakses layanan *web* untuk dilakukan penyimpanan data pada *web server* (F).
6. Laptop/PC (F) melakukan akses terhadap sistem *web* yang tersimpan pada *web server* dan juga dapat menerima data dari *server broker*.

Hasil dari pengambilan *IR code remote* AC dengan merek *sharp* yang akan digunakan sebagai perintah kontrol yang dikirimkan oleh mikrokontroler *Wemos D1 (R2)* melalui *IR Transmitter* menuju AC dapat dilihat pada tabel 3.2.

Tabel 3.2 *IR code* AC *Sharp*

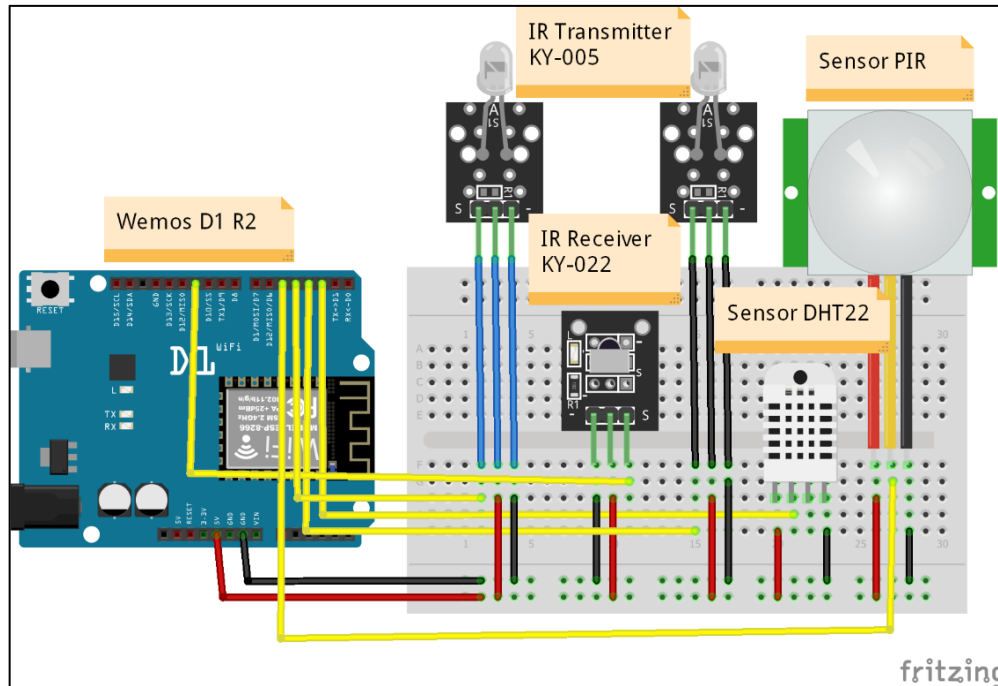
No	Tombol	Raw Data	Fungsi
1	ON	rawData[139] = {8986, 4498, 656, 1682, 656, 550, 652, 556, 650, 1684, 652, 554, 656, 552, 652, 1684, 654, 552, 654, 552, 652, 552, 656, 1682, 652, 556, 650, 556, 652, 554, 650, 556, 650, 558, 652, 554, 652,	Menghidupkan AC

		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};	
2	OFF	rawData[139] = {8982, 4500, 650, 1688, 648, 556, 624, 582, 626, 580, 652, 558, 648, 558, 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, 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};	Mematikan AC
3	16°C	rawData[139] = {8980, 4504, 646, 1690, 620, 584, 622, 586, 622, 1714, 622, 584, 648, 560, 622, 1714, 646, 562, 622, 586, 646, 558, 624, 584, 648, 558, 620, 586, 644, 560, 622, 584, 646, 560, 620, 588, 646, 562, 620, 586, 624, 582, 646, 560, 648, 1690, 648, 560, 620, 586, 644, 560, 646, 562, 622, 582, 622, 586, 618, 1718, 644, 562, 646, 1692, 620, 586, 620, 586, 646, 1690, 644, 562, 648, 20692, 620, 1716, 646, 562, 624, 584, 644, 560, 650, 1688, 650, 558, 646, 560, 624, 580, 650, 556, 646, 560, 646, 558, 622, 586, 648, 558, 624, 1712, 648, 560, 622, 582, 622, 584, 646, 560, 640, 566, 650, 556, 646, 558, 648, 560, 646, 560, 622, 584, 646, 562, 624, 582, 624,	Mengatur suhu 16°C

		584, 646, 560, 648, 558, 648, 1690, 648, 1688, 648, 558, 650};	
4	18°C	rawData[139] = {8956, 4528, 622, 1712, 648, 560, 622, 584, 646, 1692, 620, 586, 646, 560, 622, 586, 644, 560, 624, 582, 624, 1716, 646, 560, 648, 558, 644, 562, 622, 582, 624, 582, 624, 586, 620, 588, 620, 586, 622, 584, 646, 562, 622, 584, 646, 1690, 648, 556, 622, 586, 644, 562, 650, 558, 620, 584, 648, 558, 622, 1718, 618, 588, 624, 1712, 646, 562, 644, 562, 648, 1688, 644, 562, 646, 20694, 620, 586, 624, 1714, 648, 556, 648, 560, 624, 1712, 646, 558, 624, 582, 622, 584, 646, 562, 620, 586, 646, 558, 646, 562, 622, 584, 626, 1710, 650, 560, 646, 558, 624, 584, 648, 558, 622, 586, 648, 556, 650, 558, 646, 562, 622, 558, 672, 560, 622, 586, 620, 586, 648, 560, 620, 584, 650, 558, 622, 584, 646, 560, 630, 1706, 624};	Mengatur suhu 18°C
5	20°C	rawData[139] = {8982, 4502, 622, 1712, 622, 584, 648, 560, 620, 1716, 620, 586, 626, 584, 648, 558, 622, 584, 622, 584, 648, 556, 652, 1686, 650, 556, 648, 560, 650, 556, 648, 558, 626, 582, 622, 588, 620, 584, 622, 586, 620, 584, 648, 558, 650, 1686, 650, 556, 624, 584, 648, 558, 648, 560, 624, 582, 622, 586, 620, 1718, 644, 562, 624, 1714, 646, 560, 648, 560, 648, 1688, 644, 562, 624, 20716, 620, 586, 648, 1690, 648, 560, 648, 558, 622, 1714, 646, 560, 622, 584, 624, 582, 650, 558, 648, 558, 648, 560, 648, 558, 624, 584, 648, 1690, 646, 562, 648, 558, 620, 586, 650, 556, 648, 556, 652, 556, 622, 584, 622, 586, 620, 584, 622, 586, 644, 562, 622, 584, 622, 586, 644, 562, 648, 558, 620, 1712, 626, 584, 622, 1712, 650};	Mengatur suhu 20°C

3.3.2 Rangkaian Elektronika Sistem

Rangkaian elektronika sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang akan dibangun dapat dilihat pada Gambar 3.3



Gambar 3.3 Rangkaian elektronik sistem.

Gambar 3.3 merupakan rangkaian elektronika dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis yang dibuat pada penelitian ini. Berikut merupakan penjelasan dari rangkaian yang terdapat pada Gambar 3.3.

1. Sensor PIR pada rangkaian elektronika mempunyai 3 buah kabel yang terhubung dengan mikrokontroler Wemos D1 (R2). Kabel pertama dihubungkan dengan pin 5V pada mikrokontroler Wemos D1 (R2). Kabel kedua dihubungkan dengan pin digital (D5). Untuk kabel ketiga dihubungkan dengan pin GND pada mikrokontroler Wemos D1 (R2).
2. Sensor DHT22 pada rangkaian elektronik yang terdiri dari 4 kaki. Kaki pertama terhubung dengan pin 5V dari mikrokontroler Wemos D1 (R2). Untuk kaki kedua dihubungkan dengan pin digital (D2) pada mikrokontroler Wemos D1 (R2). Kaki ketiga adalah NC atau *not connected* sehingga tidak dihubungkan ke pin manapun. Kemudian kaki keempat adalah GND yang dihubungkan dengan pin GND pada mikrokontroler Wemos D1 (R2).
3. Terdapat dua sensor IR *Transmitter* KY-005 pada rangkaian elektronika dan masing-masing memiliki 3 kaki. Kaki pertama setiap sensor dihubungkan dengan pin digital (D4) dan (D3) pada mikrokontroler Wemos D1 (R2). Kaki kedua setiap sensor dihubungkan dengan pin 5V. Untuk kaki ketiga pada setiap sensor IR *Transmitter* KY-005 dihubungkan dengan pin GND pada mikrokontroler Wemos D1 (R2).

4. Sensor IR *Receiver* KY-022 pada rangkaian elektronika memiliki 3 kaki. Kaki pertama dihubungkan dengan pin GND pada mikrokontroler Wemos D1 (R2). Kaki kedua dihubungkan dengan pin 5V. Untuk kaki ketiga dihubungkan dengan pin digital (D11) pada mikrokontroler Wemos D1 (R2).

Sedangkan untuk rancangan kondisi suhu sistem akan disesuaikan dengan Standar Nasional Indonesia Termal SNI T-14-1993-03 yaitu rentang suhu sejuk-nyaman 20.5°C-22.8°C, rentang suhu nyaman-optimal 22.8°C-25.8°C dan rentang suhu hampir nyaman 25.8°C-27.1°C.

A = Sejuk Nyaman 20.5°C – 22.8°C

B = Nyaman Optimal 22.8°C – 25.8°C

C = Hampir Nyaman 25.8°C – 27.1°C

Tabel 3.3 Perencanaan kondisi sistem

No	Kondisi	Keputusan	
		Suhu	AC
1	Jika tidak ada gerakan	-	OFF
2	Jika ada gerakan AND nSuhu > batas_atas_C	16°C	ON
3	Jika ada gerakan AND nSuhu < batas_atas_C AND nSuhu > batas_bawah_C	18°C	ON
4	Jika ada gerakan AND nSuhu < batas_atas_B	20°C	ON
Target suhu yang diharapkan yaitu Nyaman Optimal		22.8°C – 25.8°C	

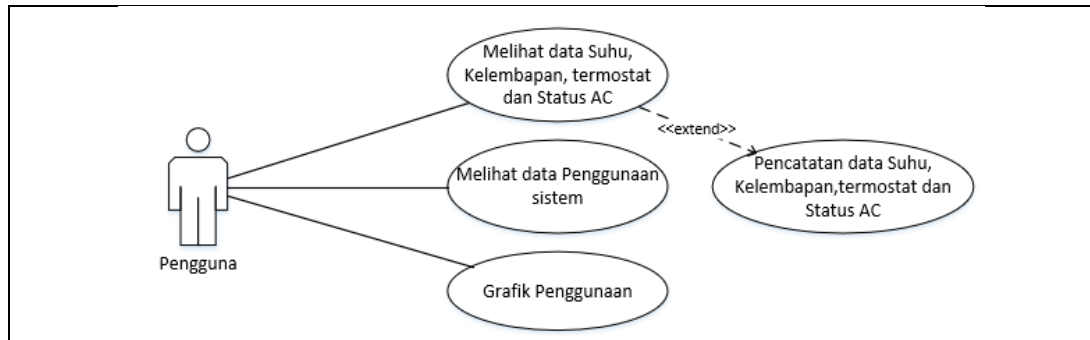
3.4 Perancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak, dilakukan perancangan sistem berbasis *web adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis. Selain merancang *web*, pada tahap ini juga akan dilakukan perancangan arsitektur MQTT sistem.

3.4.1 Sistem Monitoring Berbasis Web

Berikut ini merupakan rancangan sistem monitoring berbasis *web* manajemen penggunaan AC secara otomatis berbasis IoT. Terdiri dari rancangan *use case* diagram dan rancangan *interface* sistem

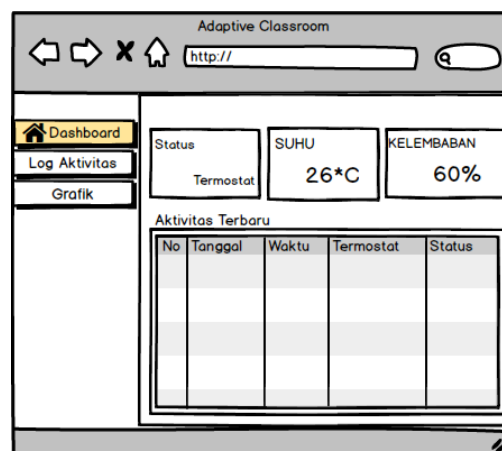
1. Rancangan Use Case Diagram



Gambar 3.4 Use case diagram sistem

Pada Gambar 3.4 merupakan *use case* diagram dari sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis. Terdapat beberapa aktivitas yang dapat dilakukan oleh operator pengguna yaitu melihat data suhu, kelembaban dan status AC secara *real time* dan dapat melihat data penggunaan sistem. Pada aktivitas melihat data suhu, kelembaban dan status AC terjadi juga proses pencatatan suhu, kelembaban termostat dan status AC secara *real time*. Sedangkan pada aktivitas melihat data penggunaan akan ditampilkan histori penggunaan AC secara periodik. Kemudian aktivitas grafik penggunaan akan digunakan untuk melihat lama waktu penggunaan AC yang ditampilkan secara periodik.

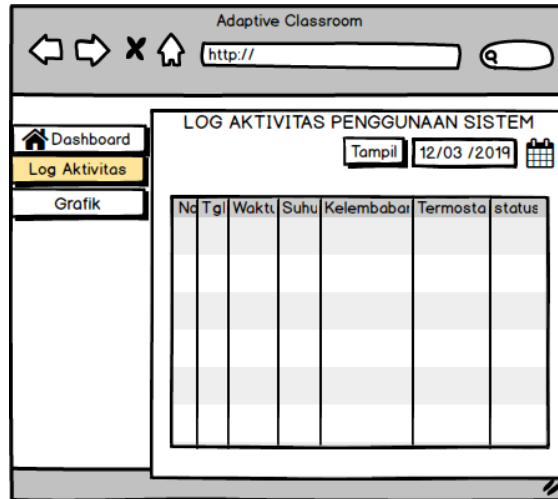
2. Rancangan interface sistem



Gambar 3.5 Tampilan halaman *dashboard*

Pada Gambar 3.5 merupakan tampilan halaman *dashboard* dari sistem *adaptive classroom* berbasis IoT manajemen penggunaan AC secara otomatis. Pada halaman

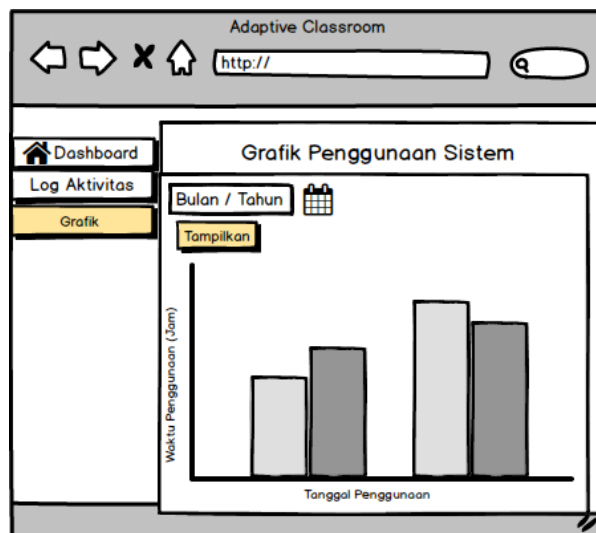
ini akan ditampilkan kondisi suhu, kelembaban, termostat dan status AC secara *real time*. Selain itu juga akan ditampilkan aktivitas terbaru penggunaan AC dalam bentuk tabel.



No	Tgl	Waktu	Suhu	Kelembaban	Termosta	status

Gambar 3.6 Tampilan halaman log aktivitas

Pada Gambar 3.6 merupakan tampilan halaman log aktivitas dari sistem *adaptive classroom* berbasis IoT manajemen penggunaan AC secara otomatis. Pada halaman ini akan ditampilkan tabel aktivitas penggunaan perangkat. Tabel aktivitas penggunaan perangkat akan berisikan tanggal, waktu, suhu, kelembaban termostat dan status AC apakah dalam kondisi menyala atau tidak.

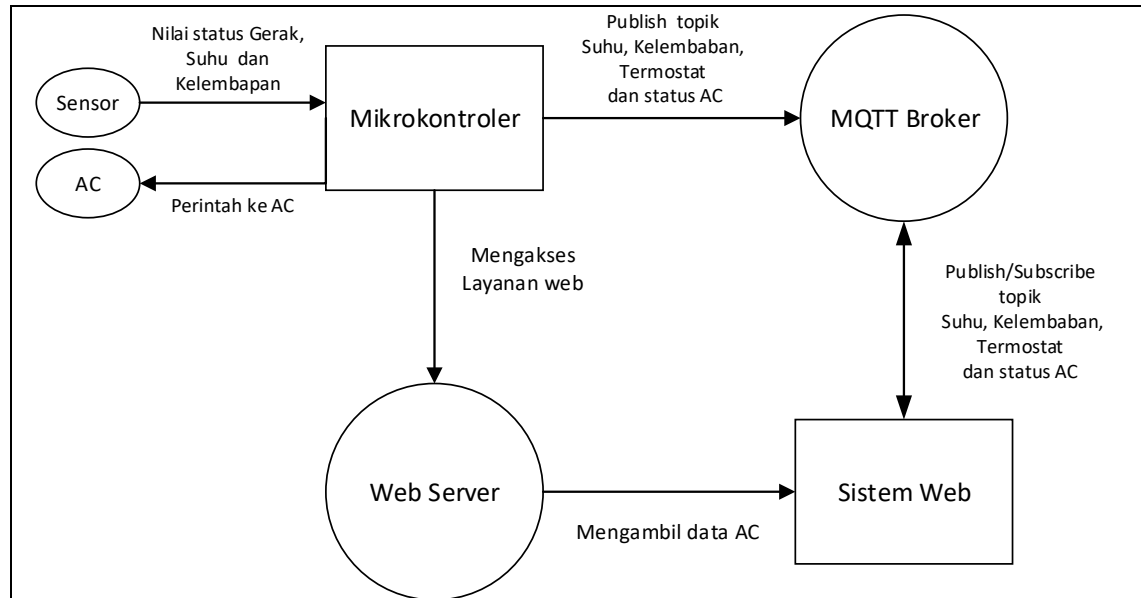


Gambar 3.7 Tampilan grafik penggunaan

Pada Gambar 3.7 merupakan tampilan halaman grafik penggunaan dari sistem *adaptive classroom* berbasis IoT manajemen penggunaan AC secara otomatis. Pada halaman ini akan ditampilkan total waktu penggunaan AC per harinya dalam bentuk grafik.

3.5 Rancangan Arsitektur MQTT Sistem

Berikut merupakan rancangan dari arsitektur MQTT sistem, pada sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis.



Gambar 3.8 Rancangan arsitektur MQTT sistem

Pada Gambar 3.8 merupakan arsitektur MQTT sistem *adaptive classroom* berbasis IoT, manajemen penggunaan AC secara otomatis. Alur proses dimulai dari mikrokontroler yang melakukan pembacaan data gerak, suhu dan kelembaban dari sensor. Kemudian mikrokontroler mem-*publish* topik suhu, kelembaban, termostat dan status AC ke MQTT *broker*. Mikrokontroler juga mengirim data penggunaan AC menuju *web server* melalui layanan *web*. Sistem *web* yang merupakan *client* dari MQTT dan *web server* mengambil data penggunaan AC yang telah dicatat pada *web server* dan juga menerima nilai sensor secara *realtime* dari MQTT *broker* dengan melakukan *subscribe* terhadap topik suhu, kelembaban, termostat dan status AC.

3.6 Implementasi Sistem

Pada tahap implementasi, akan dilakukan penyusunan konfigurasi perangkat keras dan pembuatan perangkat lunak dari rancangan yang telah dibuat sebelumnya serta menghubungkan perangkat keras dan perangkat lunak tersebut menjadi satu kesatuan sistem dengan menggunakan protokol MQTT sebagai media pengiriman data. Tahap Implementasi akan dibagi menjadi 3 proses yaitu:

1. Penyusunan konfigurasi perangkat keras

Pada tahap ini, mikrokontroler Wemos D1 (R2), sensor PIR, sensor DHT22, sensor IR *Transmitter*. Akan disusun menjadi satu rangkaian perangkat keras yang

nantinya akan digunakan untuk melakukan manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM. Proses penyusunan perangkat disesuaikan dengan rancangan pada tahap rancangan perangkat keras sistem.

2. Pembuatan perangkat lunak

Pada tahapan ini akan dilakukan pembangunan sistem berbasis *web* dengan menggunakan *framework CodeIgniter*. Sistem berbasis *web* akan digunakan untuk *monitoring* penggunaan sistem dalam melakukan manajemen penggunaan AC secara otomatis.

3. Pembangunan alur komunikasi data

Pada tahap ini akan dilakukan pembangunan komunikasi data antara sistem *web* dengan perangkat keras yang telah disusun sebelumnya. Protokol komunikasi data yang digunakan yaitu protokol MQTT dengan server broker HiveMQ.

3.7 Pengujian dan Evaluasi Sistem

Pada tahap pengujian dan evaluasi sistem, akan dilakukan pengujian terhadap sistem yang telah dibangun dan melakukan evaluasi dari pengujian yang telah dilakukan tersebut. Proses pengujian sistem *Adaptive Classroom* Berbasis IoT, Manajemen Penggunaan AC Secara Otomatis dilakukan pada ruang kelas A3-02 PSTI FT UNRAM. Pengujian sistem dibagi menjadi dua tahap. Pada tahap pertama dilakukan pengujian terhadap fungsi dari perangkat sistem manajemen penggunaan AC secara otomatis yang telah dibuat. Kemudian tahap selanjutnya adalah pengujian yang dilakukan terhadap sistem *web* yang telah terhubung dengan protokol MQTT untuk dapat dilakukan *monitoring* terhadap manajemen otomatisasi penggunaan AC yang dilakukan oleh sistem secara *real time*. Jika sistem ini sudah sesuai dengan kebutuhan yaitu dapat memanajemen penggunaan AC secara otomatis dan juga dapat menampilkan data yang sesuai pada sistem *monitoring* berbasis *web* baik berupa tabel maupun grafik maka akan dilanjutkan menuju tahap berikutnya. Sedangkan jika sistem belum berjalan sesuai dengan kebutuhan maka akan dilakukan perbaikan pada tahap perancangan perangkat keras.

3.8 Dokumentasi dan Laporan

Pada tahap dokumentasi dan laporan, hasil dari pengujian sistem akan didokumentasikan dan diambil kesimpulan berdasarkan dokumen tersebut. Kesimpulan yang telah didapatkan akan digunakan sebagai acuan untuk pengembangan selanjutnya.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Realisasi Sistem

Pada bab ini, akan membahas hasil dari penelitian yang dilakukan yaitu “Realisasi Sistem *Adaptive Classroom* Berbasis IoT, Manajemen Penggunaan *Air Conditioner* (AC) Secara Otomatis”. Realisasi yang dilakukan telah dibuat sesuai dengan perancangan yang dijabarkan pada bab sebelumnya. Selain itu, pada bab ini juga akan dibahas mengenai hasil sistem yang telah dibuat berdasarkan perancangan yang ada, melakukan pengujian keseluruhan sistem serta mengevaluasi sistem yang berjalan.

4.1.1 Realisasi Penyusunan Perangkat Keras

Realisasi penyusunan perangkat keras dari Sistem *Adaptive Classroom* Berbasis IoT, Manajemen Penggunaan *Air Conditioner* (AC) Secara Otomatis mengacu pada rancangan perangkat keras 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.

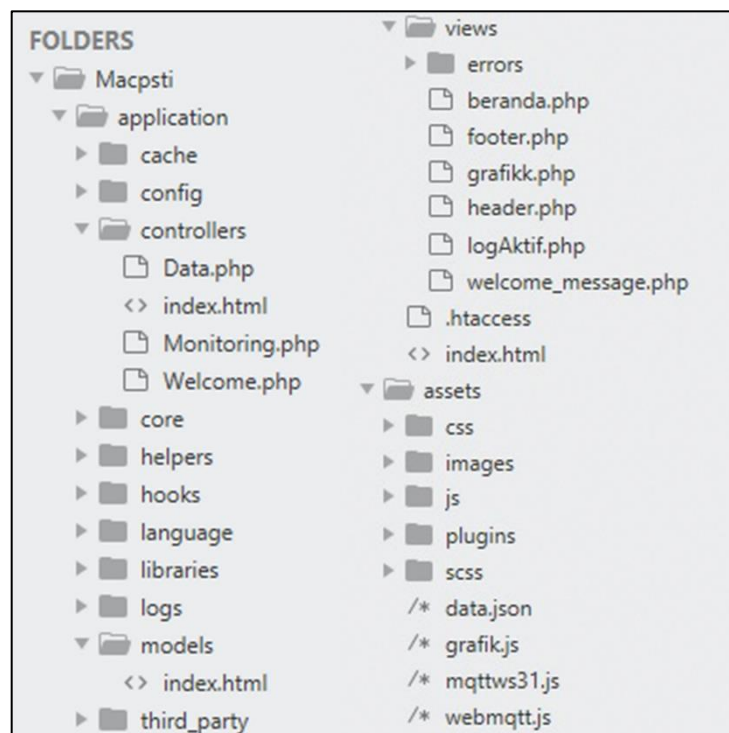
Pada Gambar 4.1 di atas terdapat lima perangkat keras yang dihubungkan menjadi sebuah perangkat untuk melakukan manajemen penggunaan AC secara otomatis pada ruang A3-02 PSTI FT UNRAM, yang terdiri dari *Wemos D1 R(2)*, dua buah IR

Transmitter, sensor DHT22, dan sensor PIR. Fungsi dari masing-masing perangkat keras tersebut adalah sebagai berikut:

1. *Wemos D1*, digunakan sebagai mikrokontroler dari sistem manajemen otomatisasi AC, yang dimana mikrokontroler ini sudah memiliki modul ESP8266 untuk terhubung ke jaringan internet.
2. *IR Transmitter*, digunakan untuk mengirim sinyal perintah ke AC berdasarkan instruksi yang telah diolah oleh mikrokontroler. *Code* instruksi yang dikirimkan disesuaikan dengan *code* instruksi yang diambil dari *remote* AC menggunakan *IR Receiver*.
3. Sensor DHT22, digunakan untuk mendeteksi suhu dan kelembaban udara pada ruangan.
4. Sensor PIR, digunakan untuk mendeteksi pancaran sinar inframerah yang dihasilkan oleh gerakan manusia.

4.1.2 Realisasi Pembangunan Web

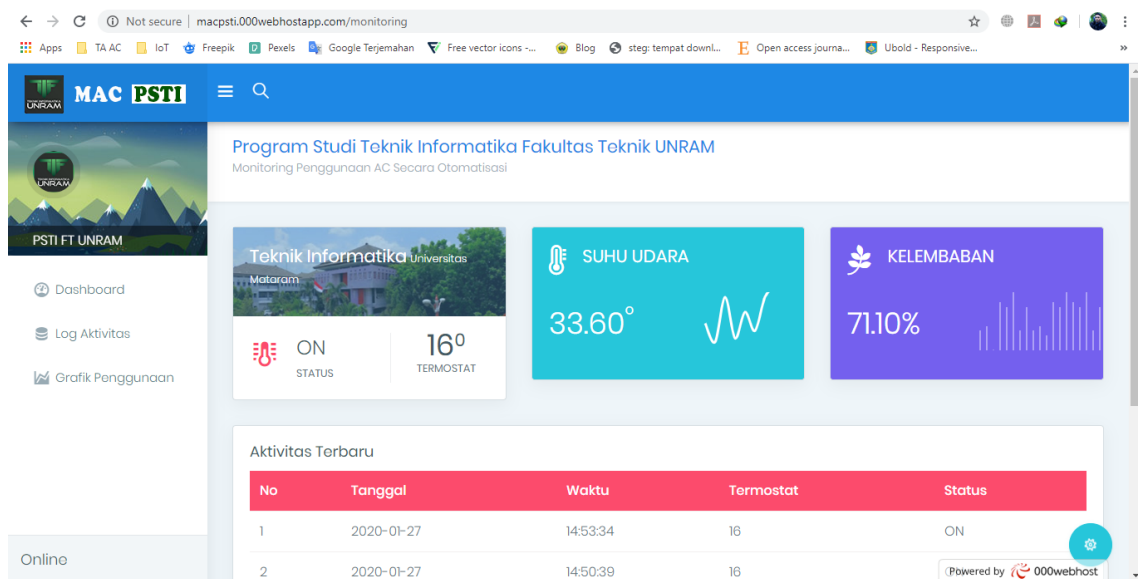
Realisasi pembangunan *web* sistem manajemen penggunaan AC secara otomatis dibuat sesuai dengan perancangan pada bab sebelumnya. Bahasa yang digunakan dalam pembangunan *web* ini adalah menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter*. Struktur folder yang terdapat pada *CodeIgniter* dapat dilihat pada Gambar 4.2 di bawah ini.



Gambar 4.2 Folder Struktur pembangunan web dengan *CodeIgniter*.

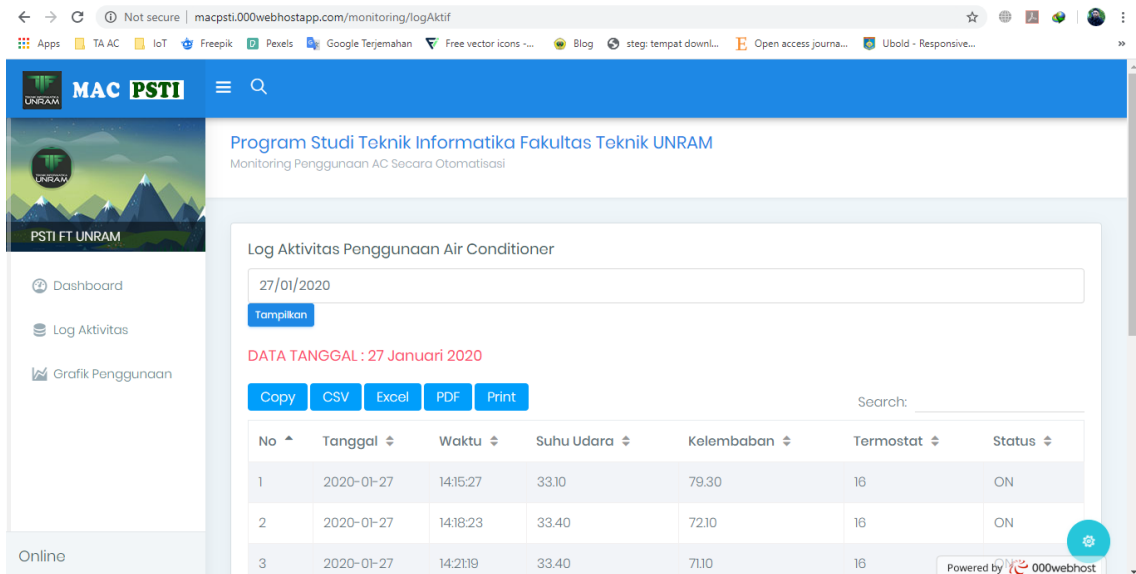
4.1.3 Realisasi *Interface* Sistem

Dalam pembuatan *web* sistem *Adaptive Classroom* berbasis IoT, Manajemen penggunaan AC Secara Otomatis digunakan bahasa PHP dengan *framework codeigniter*. Seperti yang telah dijelaskan pada *use case* diagram Gambar 3.3 pada bab sebelumnya. Terdapat satu aktor atau pengguna yang menjalankan sistem. Berikut ini adalah realisasi dari *interface* sistem *web Adaptive Classroom* Berbasis IoT, Manajemen Penggunaan AC Secara Otomatis.



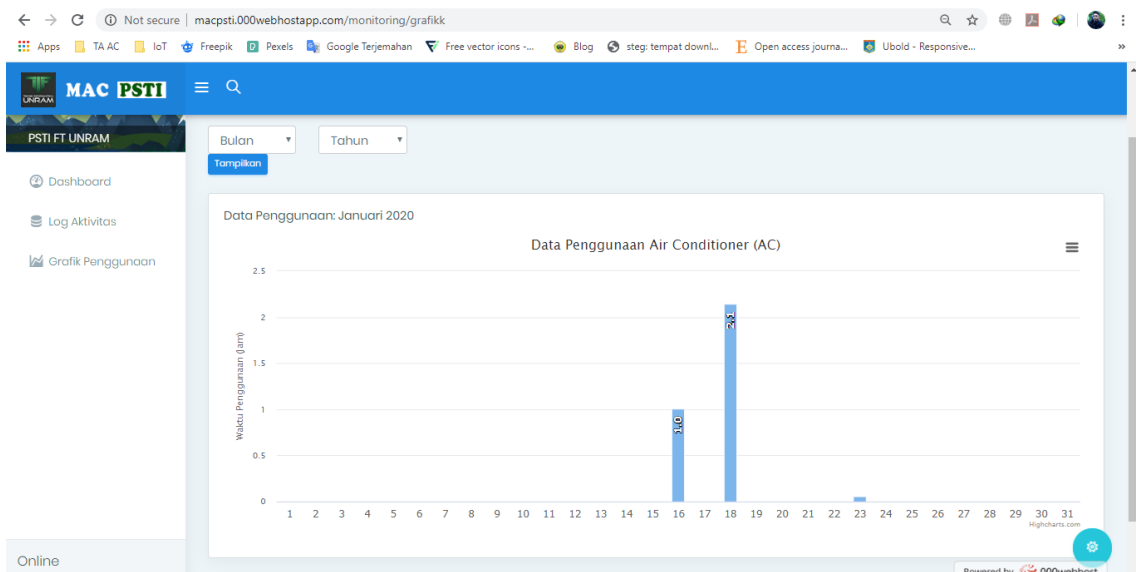
Gambar 4.3 Halaman *dashboard*

Gambar 4.3 di atas merupakan realisasi *interface* halaman *dashboard*. Halaman ini menampilkan data monitoring manajemen penggunaan AC saat ini berupa status kondisi AC, termostat AC, kondisi suhu dan kelembaban udara secara *real time*. Selain itu, pada halaman ini juga akan menampilkan data aktivitas terbaru penggunaan sistem berupa tabel.



Gambar 4.4 Halaman log aktivitas

Gambar 4.4 di atas merupakan realisasi *interface* halaman log aktivitas. Halaman ini menampilkan pencatatan data berdasarkan periodik tanggal dalam bentuk tabel. Tabel berisi data hasil monitoring manajemen penggunaan AC pada kondisi sistem dalam keadaan menyala.



Gambar 4.5 Halaman grafik penggunaan

Gambar 4.5 di atas merupakan realisasi *interface* halaman grafik penggunaan. Halaman ini menampilkan total waktu penggunaan AC yang dipresentasikan dalam bentuk grafik.

4.1.4 Realisasi Pembangunan Program pada Mikrokontroler

Realisasi pembangunan program pada mikrokontroler ini dilakukan dengan menggunakan bahasa pemrograman C dengan aplikasi Arduino IDE. Adapun program yang dibangun pada tahap ini yaitu program untuk mengambil data dari sensor, program untuk menyambungkan mikrokontroler dengan *internet* dan MQTT broker serta program untuk mengirimkan data sensor ke MQTT broker. Berikut merupakan penjelasan *source code* program pada mikrokontroler.

Listing 4.1 Script Library control application

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ESP8266HTTPClient.h>
#include "DHTesp.h"
#include <IRremoteESP8266.h>
#include <Irsend.h>
```

Script di atas merupakan *script* untuk memasukkan *library* yang akan digunakan. Adapun *library* yang digunakan pada program ini yaitu `"#include "DHTesp.h"` merupakan *library* untuk sensor DHT22, `"#include <IRremoteESP8266.h>"` dan `"#include <Irsend.h>"` merupakan *library* untuk mengirim perintah dengan menggunakan IR Transmitter, `"#include <ESP8266HTTPClient.h>"` merupakan *library* untuk HTTP Client, `"#include <ESP8266WiFi.h>"` merupakan *library* untuk modul ESP8266 pada Wemos D1 R(2), dan `"#include <PubSubClient.h>"` merupakan *library* untuk *publish* dan *subscribe* MQTT broker.

Listing 4.2 Script pendeklarasian variabel control application

```
DHTesp dht;
const uint16_t kIrLed = D6;
Irsend irsend(kIrLed);

uint16_t rawEnam[139] = {8980, 4504, 646, 1690, 620, 584, 622, 586,
622, 1714, 622, 584, 648, 560, 622, 1714, 646, 562, 622, 586, 646,
558, 624, 584, 648, 558, 620, 586, 644, 560, 622, 584, 646, 560,
620, 588, 646, 562, 620, 586, 624, 582, 646, 560, 648, 1690, 648,
560, 620, 586, 644, 560, 646, 562, 622, 582, 622, 586, 618, 1718,
644, 562, 646, 1692, 620, 586, 620, 586, 646, 1690, 644, 562,
648, 20692, 620, 1716, 646, 562, 624, 584, 644, 560, 650, 1688,
650, 558, 646, 560, 624, 580, 650, 556, 646, 560, 646, 558, 622,
586, 648, 558, 624, 1712, 648, 560, 622, 582, 622, 584, 646,
560, 640, 566, 650, 556, 646, 558, 648, 560, 646, 560, 622, 584,
646, 562, 624, 582, 624, 584, 646, 560, 648, 558, 648, 1690, 648,
1688, 648, 558, 650};

uint16_t rawOff[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, 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};

uint16_t rawLapan[139] = {8956, 4528, 622, 1712, 648, 560, 622,
584, 646, 1692, 620, 586, 646, 560, 622, 586, 644, 560, 624, 582,
624, 1716, 646, 560, 648, 558, 644, 562, 622, 582, 624, 582, 624,
586, 620, 588, 620, 586, 622, 584, 646, 562, 622, 584, 646, 1690,
648, 556, 622, 586, 644, 562, 650, 558, 620, 584, 648, 558, 622,
1718, 618, 588, 624, 1712, 646, 562, 644, 562, 648, 1688, 644,
562, 646, 20694, 620, 586, 624, 1714, 648, 556, 648, 560, 624,
1712, 646, 558, 624, 582, 622, 584, 646, 562, 620, 586, 646,
558, 646, 562, 622, 584, 626, 1710, 650, 560, 646, 558, 624, 584,
648, 558, 622, 586, 648, 556, 650, 558, 646, 562, 622, 558, 672,
560, 622, 586, 620, 586, 648, 560, 620, 584, 650, 558, 622, 584,
646, 560, 630, 1706, 624};

uint16_t rawDuapuluh[139] = {8982, 4502, 622, 1712, 622, 584, 648,
560, 620, 1716, 620, 586, 626, 584, 648, 558, 622, 584, 622, 584,
648, 556, 652, 1686, 650, 556, 648, 560, 650, 556, 648, 558, 626,
582, 622, 588, 620, 584, 622, 586, 620, 584, 648, 558, 650, 1686,
650, 556, 624, 584, 648, 558, 648, 560, 624, 582, 622, 586, 620,
1718, 644, 562, 624, 1714, 646, 560, 648, 560, 648, 1688, 644,
562, 624, 20716, 620, 586, 648, 1690, 648, 560, 648, 558, 622,
1714, 646, 560,
622, 584, 624, 582, 650, 558, 648, 558, 648, 560, 648, 558, 624,
584, 648, 1690, 646, 562, 648, 558, 620, 586, 650, 556, 648, 556,
652, 556, 622, 584, 622, 586, 620, 584, 622, 586, 644, 562, 622,
584, 622, 586, 644, 562, 648, 558, 620, 1712, 626, 584, 622,
1712, 650};

int pir = D5;
int state = LOW;
int val = 0;
int termostat;
int state_fungsi = 0;
String lembab = "";
String suhu = "";
String termo = "";
int append = 0;
String Link = "";

```

Script di atas merupakan *script* untuk mendefinisikan objek-objek *library* yang telah dimasukkan sebelumnya. Adapun objek-objek *library* yang didefinisikan yaitu *library* "DHTesp" yang didefinisikan menjadi variabel "dht", *library* "Irsend" yang didefinisikan menjadi variabel "irsend". Kemudian *script* di atas juga menginisialisasi pin yang digunakan pada mikrokontroler yaitu pin digital D6 untuk variabel "kIrLed", pin D5 untuk variabel "pir" serta mendeklarasikan variabel-variabel yang akan digunakan pada program. *Script* di atas juga mendeklarasikan RAW data yang digunakan untuk mengirim perintah melalui IR *Transmitter* yaitu "rawEnama(137)" untuk mengirim

perintah suhu 16°C, "rawOFF(137)" untuk perintah AC OFF, "rawLapan(137)" untuk perintah suhu 18°C dan "rawDuapuluh(137)" untuk perintah suhu 20°C.

Listing 4.3 Script pendeklarasian variable wifi

```
const char* ssid = "Redmi";
const char* password = "1sampai8";
const char* MQTT_server = "broker.mqtt-dashboard.com";

WiFiClient espClient;
PubSubClient client(espClient);
```

Script di atas merupakan script untuk mendeklarasikan variable-variabel wifi dan MQTT yang akan digunakan mikrokontroler Wemos D1 R(2) dengan modul ESP8266 untuk terhubung dengan jaringan *internet* dan MQTT *broker*. "ssid" merupakan variabel untuk menampung nama jaringan wifi, "password" merupakan variabel untuk menampung password jaringan wifi yang digunakan, "MQTT_server" merupakan variabel untuk menampung alamat broker MQTT yang digunakan sebagai tempat menyimpan data.

Listing 4.4 Script setup jaringan wifi

```
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Script di atas merupakan script fungsi "void setup_wifi()" untuk mengatur proses koneksi mikrokontroler Wemos D1 R(2) dengan jaringan *internet* dan MQTT yaitu dengan mengambil nilai variabel "ssid" dan "password" yang telah dideklarasikan sebelumnya.

Listing 4.5 Script fungsi reconnect

```
void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
```

```

        Serial.println("connected");
        client.publish("outTopic", "MQTT Terkoneksi");

        client.subscribe("inTopic");
    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        delay(5000);
    }
}
}
}

```

Script di atas merupakan *script* fungsi `"void reconnect()"` untuk mengatur proses koneksi kembali mikrokontroler Wemos D1 (R2) dengan jaringan *internet* dan MQTT dengan *delay* 5 detik.

Listing 4.6 *Script* fungsi *setup*

```

void setup()
{
    pinMode(BUILTIN_LED, OUTPUT);
    Serial.begin(115200);
    setup_wifi();
    client.setServer(MQTT_server, 1883);
    client.setCallback(callback);
    irsend.begin();
    #if ESP8266
        Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
    #else // ESP8266
        Serial.begin(115200, SERIAL_8N1);
    #endif // ESP8266
    dht.setup(D7, DHTesp::DHT22);

    appenddataOFF();
}

```

Script di atas merupakan *script* fungsi `"void setup()"` untuk mengatur modul-modul yang digunakan sesuai dengan *library* yang telah dimasukkan sebelumnya. `"dht.setup(D7,DHTesp::DHT22)"` merupakan pengaturan untuk sensor DHT22, `"irsend.begin()"` merupakan pengaturan untuk IR *Transmitter*, `"setup_wifi"` merupakan pengaturan untuk memanggil fungsi `"setup_wifi()"` dan `"client.setServer"` merupakan pengaturan untuk mengatur *server* yang digunakan dengan parameter nilai variabel `"MQTT_server"` dan *port* 1883. `"appenddataOFF();"` digunakan untuk menjalankan fungsi `"appenddataOFF();"` pada fungsi `"void setup()"`.

Listing 4.7 *Script* fungsi *loop*

```

void loop()
{
    if (!client.connected()) {
        reconnect();
    }
}

```

```

}
client.loop();

```

Script di atas merupakan *script* fungsi “void loop()” untuk melakukan pengulangan dalam pembacaan nilai sensor-sensor yang digunakan yang dihitung sesuai dengan *library* sensor yang telah dimasukkan sebelumnya. Fungsi ini akan berjalan jika mikrokontroler Wemos D1 R(2) sudah terhubung dengan *internet* dan MQTT broker.

Listing 4.8 *Script* pembacaan sensor DHT22.

```

Delay(dht.getMinimumSamplingPeriod());
float humidity = dht.getHumidity();
float temperature = dht.getTemperature();

lembab += humidity;
suhu += temperature;
client.publish("test/suhu", (char*) suhu.c_str());
client.publish("test/lembab", (char*) lembab.c_str());
Serial.println("===== \t");
Serial.println("Publish Data..... \t");
Serial.print("Kelembaban: \t");
Serial.println(lembab);
Serial.print("Suhu:\t");
Serial.println(suhu);

```

Script di atas merupakan bagian *script* fungsi “void loop()” yang digunakan untuk melakukan pembacaan nilai sensor suhu dan kelembaban dari variabel dan *library* yang telah dideklarasikan sebelumnya. Keluaran nilai sensor suhu ditampung dalam variabel “temperature” sedangkan nilai sensor kelembaban ditampung dalam variabel “humidity” dengan masing-masing bertipe data *float*. Kemudian nilai sensor tersebut akan di-*publish* ke MQTT broker dalam bentuk *string* dengan topik “test/suhu” dan “test/lembab”

Listing 4.9 *Script* penentuan kondisi.

```

Val = digitalRead(pir); // read pir value
if (val == HIGH && temperature > 27)
{
    if(state_fungsi != 1){
        for (int i = 0; i < 4; i++) {
            irsend.sendRaw(rawEnam, 139, 38);
            delay(2000);
        }
        termostat = 16;
    }
    state_fungsi = 1;
    statusAC= "ON";
    append += 1;
    if(append == 40){
        appenddata();
        append = 0;
    }
}
else if (val == HIGH && temperature < 27 && temperature > 25.80)
{

```

```

        if(state_fungsi != 2){
            for (int i = 0; i < 4; i++) {
                irsend.sendRaw(rawLapan, 139, 38); //send code 18
                delay(1000);
            }
            termostat = 18;
        }
        state_fungsi = 2;
        statusAC= "ON";
        append += 1;
        if(append == 40){
            appenddata();
            append = 0;
        }
    }
    else if (val == HIGH && temperature < 25.80)
    {
        if(state_fungsi != 3){
            for (int i = 0; i < 4; i++) {
                irsend.sendRaw(rawDuapuluh, 139, 38); //send code 20
                delay(1000);
            }

            termostat = 20;
        }
        state_fungsi = 3;
        statusAC= "ON";
        append += 1;
        if(append == 40){
            appenddata();
            append = 0;
        }
    }
    else if (val == LOW){
        if(state_fungsi != 0){
            for (int i = 1; i < 4; i++){
                irsend.sendRaw(rawOff, 139, 38); //send Mati
                delay(1000);
            }
            termostat = 0;
        }
        state_fungsi = 0;
        statusAC= "OFF";
        append += 1;
        if(append == 40){
            appenddata();
            append = 0;
        }
    }
    }
    delay(2000);

```

Script di atas merupakan bagian *script* fungsi "void loop()" yang digunakan untuk menghitung dan menentukan kondisi yang akan dikirimkan ke AC melalui "irsend.sendRaw" berdasarkan nilai dari sensor PIR dan sensor DHT22 yang telah didapatkan sebelumnya. Perhitungan dan penentuan nilai variabel tersebut menggunakan kondisi *if else*.

Listing 4.10 *Script publish data*

```
String stateFungsi = "";
stateFungsi += state_fungsi;

client.publish("test/fungsi", (char*) stateFungsi.c_str());
termo += termostat;
client.publish("test/termostat", (char*) termo.c_str());
    Serial.print("fungsi:\t");
    Serial.println(stateFungsi);
    Serial.print("termostat: \t");
    Serial.println(termo);
suhu = "";
termo = "";
lembab = "";
    Serial.print( "append count \t");
    Serial.println(append);
```

Script di atas merupakan bagian *script* fungsi "void loop()" yang digunakan untuk melakukan *publish* data hasil perhitungan dan penentuan kondisi pada *script* sebelumnya ke MQTT broker dalam bentuk *string*. State fungsi yang dieksekusi akan di-*publish* dengan topik "test/fungsi" yang digunakan untuk mengetahui status kondisi AC apakah dalam state fungsi aktif atau tidak.

Listing 4.11 *Script fungsi simpan data*

```
void appenddata() {
    HTTPClient http;
    Link = "http://macpsti.000webhostapp.com/data/appenddata/" + suhu
+ "/" + lembab + "/" + termostat + "/" + statusAC;
    http.begin(Link);
    int httpCode = http.GET();
    delay(10);
    http.end();
}
```

Script di atas merupakan fungsi "appenddata()" yang digunakan untuk melakukan penyimpanan data. Data yang disimpan berupa data suhu, kelembaban, termostat dan status AC. Fungsi ini akan dijalankan pada fungsi "void loop()" berdasarkan hasil penentuan kondisi dari *if-else*.

Listing 4.12 *Script fungsi simpan data off*

```
void appenddataOFF() {
    HTTPClient http;
    Link = "http://macpsti.000webhostapp.com/data/appendOFF/";
    http.begin(Link);
    int httpCode = http.GET();
    delay(10);
    http.end();
}
```

Script di atas merupakan fungsi "appenddataOFF()" yang digunakan untuk melakukan penyimpanan data. Data yang disimpan berupa data suhu, kelembaban, termostat dan status AC dalam kondisi OFF. Fungsi ini akan dijalankan pada fungsi

“void setup()” untuk mengkondisikan status AC dalam kondisi OFF ketika pertama kali sistem dijalankan. Hal ini dilakukan untuk memudahkan sistem dalam mengkalkulasi lama penggunaan AC perharinya yang nantinya akan ditampilkan dalam bentuk grafik penggunaan sistem.

4.1.5 Realisasi Pembangunan Arsitektur Komunikasi Data MQTT

Pada realisasi pembangunan arsitektur komunikasi data dengan MQTT ini menggunakan dua *library* yang disematkan pada sistem *web* agar dapat terhubung dan melakukan aksi pada server broker. *Library* yang digunakan yakni berkas *jQuery* dan *mqttws31.js*. Berikut merupakan *script* yang digunakan dalam realisasi komunikasi dengan MQTT yang tersimpan dalam berkas *webmqtt.js*.

Listing 4.13 *Script* fungsi koneksi broker

```
var 40emb;
var 40embab;

var client = new Messaging.Client("broker.mqttdashboard.com", 8000,
"Macpsti" + parseInt(Math.random() * 100, 10));
```

Script di atas digunakan untuk terhubung ke MQTT broker HiveMQ dengan random id *client*. *Script* di atas juga mendeklarasikan variabel global “suhu” dan “40embab” untuk menampung nilai suhu dan kelembaban.

Listing 4.14 *Script* fungsi *lost connection*

```
client.onConnectionLost = function (responseObject) {
    $('#status').empty();
    $('#status').append('<a href="" class="link" data-
toggle="tooltip" title="Terputus">Terputus<i class=""></i></a>');
    client.connect(options);
};
```

Script di atas merupakan fungsi yang akan dijalankan jika terjadi kehilangan koneksi dengan server broker. Pada fungsi ini, sistem *web* akan mengubah status koneksi menjadi “Terputus” kemudian menjalankan fungsi untuk menghubungkan kembali ke server broker.

Listing 4.15 *Script* fungsi penerimaan pesan

```
client.onMessageArrived = function (message) {
    if(message.destinationName == 'test/suhu'){
        $('#suhu').empty();
        $('#suhu').append('<font class="display-6 text-
white">'+message.payloadString+'<sup>°</sup></font>');
        40emb = message.payloadString;
        $('#suhuu').empty();
        $('#suhuu').append(40emb);
    }
    else if(message.destinationName == 'test/termostat'){
        $('#termostat').empty();
        $('#termostat').append('<h1 class="font m-b-
0">'+message.payloadString+'<sup>0</sup></h1>');
    }
```

```

else if(message.destinationName == 'test/lembab'){
    $('#kelembaban').empty();
    $('#kelembaban').append('<font class="display-6 text-
white">'+message.payloadString+'<sup></sup>%</font>');
    4lembab = message.payloadString;
    $('#suhuu').empty();
    $('#suhuu').append(4lembab);}
else if(message.destinationName == 'test/fungsi'){
    if(message.payloadString == '0'){
        $('#fungsi').empty();
        $('#fungsi').append('<div class="display-6 text-danger "><i
class="fa fa-power-off"></i></div>');
        $('#off').empty();
        $('#off').append('<h2 class="font-light text m-b-
0">OFF</h2>');}
    else{
        $('#fungsi').empty();
        $('#fungsi').append('<div class="display-6 text-danger "><i
class="mdi mdi-thermometer-lines"></i></div>');
        $('#off').empty();
        $('#off').append('<h2 class="font-light text m-b-0">ON</h2>');
    }
}
};

```

Script di atas merupakan fungsi yang digunakan untuk menerima pesan yang dikirimkan melalui server broker oleh mikrokontroler. Dalam fungsi tersebut, terdapat kondisi *if else* yang digunakan untuk melakukan suatu aksi terhadap sistem *web*. Aksi yang dilakukan yakni mengubah informasi pada sistem *web* untuk disajikan kepada pengguna sesuai kondisi yang diterima.

Listing 4.16 *Script* deklarasi variabel *options*

```

var options = {
    timeout: 3,
    onSuccess: function () {
        client.subscribe('test/suhu', {qos: 2});
        client.subscribe('test/lembab', {qos: 2});
        client.subscribe('test/fungsi', {qos: 2});
        client.subscribe('test/termostat', {qos: 2});
        $('#status').empty();
        $('#status').append('<a href="" class="link" data-
toggle="tooltip" title="Online">Online<i class=""></i></a>');
    },
    onFailure: function (message) {
        $('#status').empty();
        $('#status').append('<a href="" class="link" data-
toggle="tooltip" title="Offline">Offline<i class=""></i></a>');
        client.connect(options);
    }
};

```

Script di atas merupakan pendeklarasian variabel *options* yang di dalamnya terdapat dua fungsi yaitu *onSuccess* dan *onFailure*. Pada fungsi *onSuccess*, jika koneksi berhasil dilakukan dengan server broker maka sistem *web* akan melakukan *subscribe* dan mengubah status koneksi pada *web* menjadi *Online*. Sedangkan pada

fungsi `onFailure`, sistem *web* akan mengubah status koneksi menjadi “Offline” kemudian menjalankan fungsi `client.connect(options)` untuk menghubungkan kembali sistem *web* dengan server broker

4.2 Pengujian dan Evaluasi Sistem

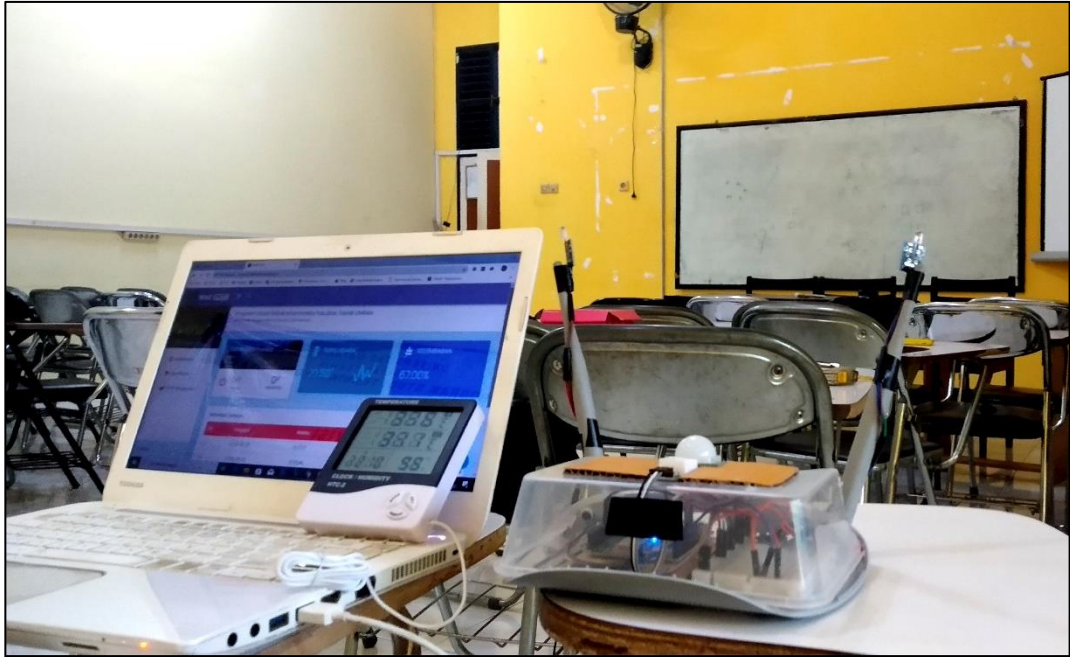
Pengujian sistem merupakan proses pengujian perangkat keras dan perangkat lunak sistem untuk mengetahui apakah sistem sudah berjalan sesuai dengan yang diinginkan peneliti. Kemudian proses evaluasi sistem dilakukan jika sistem belum berjalan sesuai dengan yang diharapkan peneliti. Pengujian perangkat keras sistem dilakukan dengan melakukan percobaan terhadap perangkat keras. Kemudian pengujian perangkat lunak sistem dilakukan dengan melakukan pengujian terhadap fungsi dari sistem monitoring manajemen penggunaan AC secara otomatis berbasis *web*. Metode yang digunakan dalam pengujian sistem ini adalah dengan menggunakan metode *black box*. Kemudian untuk pengujian keseluruhan sistem akan dilakukan di ruang A3-02 PSTI FT UNRAM.

4.2.1 Hasil Pengujian Perangkat Keras Sistem

Proses pengujian perangkat keras sistem manajemen otomatisasi AC dengan metode pengujian *black box*, yaitu menguji perangkat keras dari segi fungsionalnya saja. Pengujian perangkat keras ini digunakan untuk mengetahui apakah fungsi dan keluaran dari perangkat keras yang digunakan sudah sesuai atau tidak. Berikut merupakan hasil pengujian dari perangkat keras yang digunakan

1. Pengujian Sensor DHT22

Pengujian sensor DHT22 ditempatkan pada Ruang A3-02 untuk mengukur suhu dan kelembaban udara. Hasil pembacaan sensor DHT22 kemudian dibandingkan dengan termometer dan *hygrometer* digital. Pengujian sensor DHT22 dapat dilihat pada tabel 4.1



Gambar 4.6. Pengujian sensor DHT22

Hasil pengujian sensor DHT22 dapat dilihat pada Tabel 4.1 dan Tabel 4.2:

Tabel 4.1 Hasil pengujian suhu udara sensor DHT22

No	Termometer ($^{\circ}\text{C}$)	Sensor DHT22 ($^{\circ}\text{C}$)	Selisih	Galat Relatif (%)
1.	32.0	32.5	0.5	1.56
2.	31.5	32.0	0.5	1.58
3.	30.7	31.1	0.4	1.30
4.	29.8	30.1	0.3	1.00
5.	29.1	29.6	0.5	1.71
Rata – Rata				1.43

Pada Tabel 4.1 menunjukkan bahwa sensor telah bekerja dengan baik, hasil pengujian pengukuran suhu udara sensor DHT22 dibandingkan dengan termometer digital dengan nilai rata – rata galat 1.43%.

Tabel 4.2 Hasil Pengujian Kelembaban Udara Sensor DHT22

No	Hygrometer (%)	Sensor DHT22 (%)	Selisih	Galat Relatif (%)
1	76	77	1	1.31
2	68	70	2	2.94
3	60	62	2	3.33
4	58	60	2	3.44
5	55	56	1	1.81
Rata – Rata				2.56

Pada Tabel 4.2 menunjukkan bahwa sensor telah bekerja dengan baik, hasil pengujian pengukuran kelembaban udara sensor DHT22 dibandingkan dengan hygrometer digital dengan nilai rata – rata galat sebesar 2.56 %.

2. Pengujian Sensor PIR

Pengujian sensor PIR dilakukan pada Ruang A3-02 untuk mengukur jangkauan efektif sensor dalam mendeteksi sinyal gerakan manusia. Pengujian sensor PIR dapat dilihat pada tabel 4.3

Tabel 4.3 Hasil pengujian sensor PIR

Sudut	Jarak (Meter)	Kondisi Sensor
0°	1	<i>High</i>
	2	<i>High</i>
	3	<i>High</i>
	4	<i>High</i>
	5	<i>High</i>
	6	<i>Low</i>
30°	1	<i>High</i>
	2	<i>High</i>
	3	<i>High</i>
	4	<i>High</i>
	5	<i>High</i>
	6	<i>Low</i>
60°	1	<i>High</i>
	2	<i>High</i>
	3	<i>High</i>
	4	<i>High</i>
	5	<i>High</i>
	6	<i>Low</i>
80°	1	<i>Low</i>
	2	<i>Low</i>
	3	<i>Low</i>
	4	<i>Low</i>
	5	<i>Low</i>
	6	<i>Low</i>

Pada Tabel 4.3 menunjukkan bahwa sensor PIR telah bekerja dengan baik dan mampu mendeteksi gerakan dengan optimal pada jarak maksimal 5 meter dengan sudut maksimal 60° searah jarum jam dan 60° berlawanan jarum jam.

3. Pengujian IR Transmitter

Pengujian IR Transmitter dilakukan pada Ruang A3-02 untuk mengukur jangkauan efektif IR Transmitter dalam mengirimkan sinyal instruksi ke AC. Pengujian IR Transmitter dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil pengujian IR *Transmitter*

Jarak (Meter)	Respon AC(Ada/Tidak)
1	Ada
2	Ada
3	Ada
4	Ada
5	Ada
6	Tidak

Pada Tabel 4.4 menunjukkan bahwa IR *Transmitter* telah bekerja dengan baik dan mampu mengirim sinyal instruksi ke AC dengan jarak maksimal yaitu sepanjang 5 meter.

4.2.2 Hasil Pengujian Sistem Web

Proses pengujian *web* sistem manajemen penggunaan AC secara otomatis ini dilakukan dengan metode pengujian *black box*. Pengujian ini dilakukan untuk menguji fungsi atau fitur yang ada di dalam *web*. Berikut merupakan hasil pengujian fungsi-fungsi yang ada di *web* sistem manajemen penggunaan AC secara otomatis.

1. Pengujian Fungsi Log Aktivitas

Pengujian fungsi Log Aktivitas ini dilakukan untuk mengetahui apakah fungsi ini sudah berjalan dengan baik dan benar saat *user* atau pengguna mengakses fungsi Log Aktivitas sistem. Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa fungsi Log Aktivitas sudah berjalan dengan baik dan benar. Tabel 4.5 di bawah ini merupakan hasil pengujian yang telah dilakukan.

Tabel 4.5 Pengujian fungsi Log Aktivitas

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	<i>Form</i> pilih tanggal diisi dengan tanggal yang tidak memiliki data di <i>file</i> data	Fungsi menampilkan data kosong pada tabel fungsi log aktivitas	Sesuai	Valid
2	<i>Form</i> pilih tanggal diisi dengan tanggal yang memiliki data di <i>file</i> data	Fungsi menampilkan data Log Aktivitas pada tabel fungsi log aktivitas	Sesuai	Valid

2. Pengujian Fungsi Grafik Penggunaan

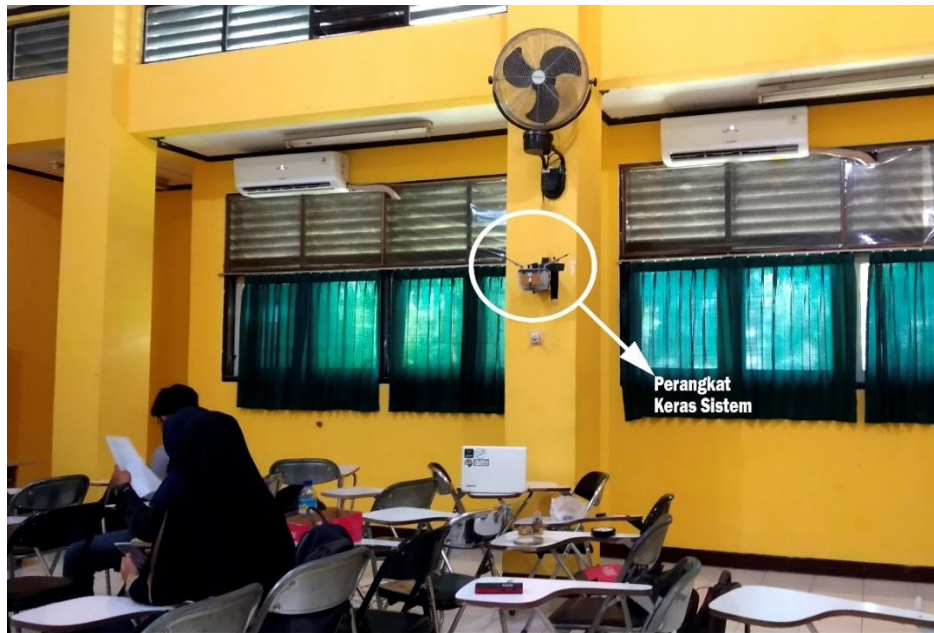
Pengujian fungsi Grafik Penggunaan ini dilakukan untuk mengetahui apakah fungsi ini sudah berjalan dengan baik dan benar saat *user* atau pengguna mengakses fungsi grafik penggunaan sistem. Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa fungsi grafik penggunaan sudah berjalan dengan baik dan benar. Tabel 4.6 di bawah ini merupakan hasil pengujian yang telah dilakukan.

Tabel 4.6 Pengujian fungsi Grafik Penggunaan

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Form pilih tanggal diisi dengan tanggal yang tidak memiliki data di <i>file</i> data	Fungsi menampilkan data kosong pada fungsi Grafik Penggunaan	Sesuai	Valid
2	Form pilih tanggal diisi dengan tanggal yang memiliki data di <i>file</i> data	Fungsi menampilkan data Grafik Penggunaan pada fungsi Grafik Penggunaan	Sesuai	Valid

4.2.3 Hasil Pengujian Fungsi Keseluruhan Sistem

Dilakukan beberapa skenario pengujian untuk mengetahui jika sistem dapat berjalan dengan baik atau tidak. Skenario yang diuji yaitu skenario dengan ruangan yang berisi mahasiswa dan skenario dengan ruangan kosong. Pengujian dilakukan pada ruang kelas A3-02 PSTI FT UNRAM yang sudah terpasang 2 buah AC. Pengujian keseluruhan sistem dilakukan pada tanggal 29 Januari 2020 mulai dari pukul 10.00 WITA sampai pukul 15.00 WITA. Perangkat manajemen otomatisasi AC di letakkan pada lokasi yang strategis yaitu sesuai dengan jarak optimal pendeteksian sensor seperti pada Gambar 4.7



Gambar 4.7 Lokasi penempatan perangkat keras sistem

Sensor-sensor akan melakukan pembacaan kondisi ruangan, suhu udara dan kelembaban udara. Kemudian berdasarkan data pembacaan sensor akan disesuaikan dengan kondisi atau nilai parameter yang menghasilkan keputusan yang akan dikirimkan melalui IR *Transmitter* ke AC. Seluruh data pembacaan sensor dan status AC akan dikirim dan ditampilkan ke sistem *web*.

No	Tanggal	Waktu	Suhu Udara	Kelembaban	Termostat	Status
40	2020-01-29	14:21:10	25.20	67.10	20	ON
39	2020-01-29	14:16:43	25.30	67.00	20	ON
38	2020-01-29	14:12:06	25.60	66.20	20	ON
37	2020-01-29	14:07:36	25.90	65.40	18	ON
36	2020-01-29	14:03:02	26.60	64.90	18	ON
35	2020-01-29	13:58:47	27.30	64.20	16	ON
34	2020-01-29	13:54:07	28.20	62.10	16	ON
33	2020-01-29	13:49:57	29.10	62.90	0	OFF
32	2020-01-29	13:45:16	28.40	61.70	0	OFF

Gambar 4.8 Log aktivitas sistem

Pada Gambar 4.8 merupakan hasil pengujian berupa data tanggal, waktu, nilai suhu udara, kelembaban, termostat dan status AC. Pada tahap pengujian ini, kondisi sistem disesuaikan dengan perancangan kondisi pada Tabel 3.2.

Hasil dari pengujian skenario pertama, yaitu ruangan berisi mahasiswa, ditunjukkan oleh data ke-34 sampai dengan 40. Dapat dilihat pada data ke-34 sampai dengan 35, bahwa status AC telah menunjukkan keputusan yang benar, yaitu AC dalam keadaan Aktif dengan termostat 16°C karena nilai suhu udara berada di atas 27.10°C. Seiring berjalan waktu, suhu udara dalam ruangan mengalami penurunan. Hal ini dapat dilihat pada data ke-36 sampai dengan 37, status AC dalam keadaan Aktif dengan termostat 18°C karena kondisi suhu berada diantara 27.10°C - 25.80°C. Selanjutnya pada data ke-38 sampai dengan 40 juga menunjukkan keputusan yang benar, yaitu status AC dalam keadaan Aktif dengan termostat 20°C karena kondisi suhu berada di bawah 25.80°C.

Kemudian untuk pengujian skenario kedua, yaitu ruang kosong, ditunjukkan pada data ke-33. Sistem akan mematikan AC secara otomatis, jika sudah tidak mendeteksi adanya gerakan manusia di dalam ruangan selama selang waktu 5 menit. Dapat dilihat pada data ke-33, status AC dikondisikan menjadi Tidak Aktif dan termostat AC diubah menjadi 0°C yang menandakan tidak adanya gerakan manusia yang terdeteksi.

Berdasarkan pengujian 2 skenario, yaitu skenario ruangan berisi mahasiswa dan skenario ruang kosong, yang telah dilakukan, perangkat IoT yang dibuat dapat melakukan manajemen penggunaan AC secara otomatis. Dengan demikian, ruang kelas yang dipasangkan perangkat IoT ini dapat dikatakan sebagai *adaptive classroom* karena dapat melakukan penyesuaian suhu secara otomatis.

4.2.4 Hasil Pengujian dengan Metode Mean Opinion Score (MOS)

Pengujian dengan menggunakan metode MOS ini dilakukan dengan responden dari mahasiswa PSTI FT UNRAM. Tujuan dari pengujian ini adalah mengetahui bagaimana kualitas sistem dilihat dari sisi pengguna. Adapun pertanyaan kuesioner yang diberikan adalah sebagai berikut:

1. Sistem manajemen otomatisasi AC berbasis IoT dapat berjalan dengan baik dan sesuai dengan harapan?
2. *Interface* sistem *web* manajemen otomatisasi AC berbasis *web* ini mudah dipahami dan digunakan (*user friendly*) ?
3. Sistem monitoring berbasis *web* pada Sistem manajemen otomatisasi AC berbasis IoT dapat memberikan informasi mengenai kondisi lingkungan suhu dan kelembaban udara serta status kondisi AC di dalam ruang kelas A3-02 PSTI FT UNRAM?

4. Fungsi filter data grafik dan log aktivitas penggunaan mudah dipahami dan digunakan?
5. Apakah sistem ini dapat membantu atau memudahkan manajemen penggunaan AC secara otomatis di dalam ruang kelas A3-02 PSTI FT UNRAM ?

Hasil pengujian parameter MOS yang diperoleh dari 20 responden dapat dilihat pada Tabel 4.7.

Tabel 4.7 Pengujian parameter MOS

No	Pertanyaan	SS(5)	S(4)	TT(3)	TS(2)	STS(1)	Total	Mean pi
1	Pertanyaan 1	10	10	0	0	0	20	4.5
2	Pertanyaan 2	16	4	0	0	0	20	4.8
3	Pertanyaan 3	7	12	1	0	0	20	4.3
4	Pertanyaan 4	11	9	0	0	0	20	4.55
5	Pertanyaan 5	14	6	0	0	0	20	4.7
Sub Total		58	41	1	0	0	100	22.85
MOS (Mean Opinion Score)								4.57

Berikut ini merupakan rumus yang digunakan dalam menghitung skor rata-rata dari bobot nilai yang didapat [19]:

$$Mean\ pi = \frac{\sum Si . Bi}{n}$$

Dimana:

mean pi : rata-rata skor setiap atribut pertanyaan

Si : jumlah responden yang memilih setiap atribut jawaban

Bi : bobot setiap atribut pertanyaan

n : jumlah responden

Untuk mendapatkan nilai MOS (*Mean Opinion Score*) berdasarkan skor rata-rata yang didapat, dapat digunakan rumus berikut:

$$MOS = \frac{\sum_{i=1}^k Mean\ pi}{k}$$

Dimana:

MOS : total skor rata-rata seluruh atribut pertanyaan

K : jumlah atribut pertanyaan

Hasil pengujian MOS pada Tabel 4.7 menunjukkan bahwa nilai MOS yang diperoleh yaitu sebesar 4.57. Nilai 4.57 dari skala 5 ini menunjukkan bahwa sistem yang dibangun sudah sangat baik sudah berjalan sesuai dengan apa yang diharapkan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian dan pengujian yang telah dilakukan, maka kesimpulan yang dapat diambil adalah sebagai berikut.

1. Hasil pengukuran suhu udara menggunakan sensor DHT22 dibandingkan dengan termometer digital menghasilkan nilai rata-rata galat 1.43 %. Sedangkan hasil pengukuran kelembaban udara sensor DHT22 dibandingkan dengan *hygrometer* digital dengan nilai rata – rata galat sebesar 2.56 %.
2. Sensor PIR telah bekerja dengan baik dan mampu mendeteksi gerakan manusia dengan optimal pada jarak maksimal 5 meter dengan sudut maksimal 60° searah jarum jam dan 60° berlawanan arah jarum jam. Sedangkan IR *Transmitter* yang digunakan telah bekerja dengan baik dan mampu mengirim sinyal instruksi ke AC dengan jarak maksimal yaitu sepanjang 5 meter.
3. *Web* sistem manajemen penggunaan AC secara otomatis berbasis IoT sudah dapat menampilkan data suhu udara, kelembaban, status AC dan termostat yang didapatkan dari rangkaian elektronik secara *real-time*, dalam bentuk tabel dan grafik.
4. Berdasarkan dua skenario pengujian yaitu skenario ruangan berisi mahasiswa dan skenario ruang kosong, yang telah dilakukan, perangkat IoT yang dibuat dapat melakukan manajemen penggunaan AC secara otomatis. Dengan demikian, ruang kelas yang dipasangkan perangkat IoT ini dapat dikatakan sebagai *adaptive classroom* karena dapat melakukan penyesuaian suhu secara otomatis.
5. Dari pengujian MOS yang telah dilakukan diperoleh nilai sebesar 4.57 dari skala 5 yang menunjukkan bahwa sistem telah layak digunakan dan termasuk dalam kategori sangat baik.

5.2 Saran

Adapun saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Diharapkan sistem yang dibangun dapat melakukan input IR *code* dari sistem *web* agar sistem ini dapat diterapkan pada semua jenis AC.
2. Diharapkan sistem yang dibangun dapat dikembangkan dengan menambahkan sensor PIR agar dapat meningkatkan akurasi serta dapat menjangkau dan mendeteksi pergerakan manusia di semua sudut ruangan.

DAFTAR PUSTAKA

- [1] M. Attar, B. Hamzah, and M. R. Rahim, “Kenyamanan Termal Ruang Kuliah dengan Pengkondisian Udara,” Universitas Hasanuddin Makassar, 2014.
- [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. Artono and F. Susanto, “Wireless Smart Home System Menggunakan Internet of Things,” *J. Teknol. Inform. dan Terap.*, vol. 05, no. 01, pp. 17–24, 2017.
- [4] 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.
- [5] 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, pp. 201–205, 2018.
- [6] I. G. M. N. Desnanjaya, I. A. D. Giriantri, and R. S. Hartati, “Rancang Bangun Sistem Control Air Conditioning Automatis Berbasis Passive Infrared Receiver,” *Pros. Conf. Smart-Green Technol. Electr. Inf. Syst.*, no. C–010, pp. 165–171, 2013.
- [7] H. Izzatul Islam *et al.*, “Sistem Kendali Suhu Dan Pemantauan Kelembaban Udara Ruangan Berbasis Arduino Uno Dengan Menggunakan Sensor Dht22 Dan Passive Infrared (Pir),” *Pros. Semin. Nasioal Fis. (E-Journal) SNFT2016*, vol. V, 2016.
- [8] D. A. Kurniawan, “Pengendalian Air Conditioner Dari Jarak Jauh,” Universitas Muhammadiyah Surakarta, 2018.
- [9] D. T. Istiningrum *et al.*, “Kajian Kenyamanan Termal Ruang Kuliah pada Gedung Sekolah C Lantai 2 Politeknik Negeri Semarang,” *Wahana Tek. SIPIL*, vol. 22, pp. 1–16, 2017.
- [10] B. R. U. Putri, I. W. A. Arimbawa, and F. Bimantoro, “Sistem Presensi Siswa Berbasis Internet of Things Menggunakan Sensor Sidik Jari pada SMK Perhotelan 45 Mataram,” *JTIKA*, vol. 1, no. 2, pp. 224–232, 2019.
- [11] D. A. R. Fajrika hadnis Putra, Kemas Muslim Lhaksmana, “Aplikasi IoT untuk Rumah Pintar dengan Fitur Prediksi Cuaca,” *J. e-Proceeding Eng.*, vol. 5, no. 1, pp. 1746–1760, 2018.
- [12] Arfandi Isnaeni, “Rancang Bangun Smarthome Menggunakan Chat Bot Telegram Berbasis Arduino,” Universitas Islam Negeri Alauddin Makassar, 2018.

- [13] Rini Herlina Dakhi, “Sistem Pemantau Ruang Jarak Jauh Menggunakan Sensor PIR (Passive Infrared) Berbasis Atmega 8535,” Universitas Sumatera Utara, 2017.
- [14] W. Febriani, “Alat Pendeteksi Maling pada Rumah Menggunakan Sensor Passive Infrared Berbasis Arduino,” Universitas Sumatera Utara, 2017.
- [15] Y. A. K. Utama, “Perbandingan Kualitas Antar Sensor Suhu dengan menggunakan Arduino Pro Perb Mini,” *e-Jurnal Nar.*, vol. 2, 2016.
- [16] Yusniati, “Penggunaan Sensor Infrared Switching Pada Motor DC Satu Phasa,” *J. Elektr. Technol.*, vol. 3, pp. 90–96, 2018.
- [17] I. M. A. Suyadnya, I. G. Agung, and P. Raka, “Sistem Monitoring Penetasan Telur Penyu Menggunakan Mikrokontroler NodeMCU ESP8266 dan Protokol MQTT dengan Notifikasi Berbasis Telegram Messenger,” *J-COSINE*, vol. 2, no. 2, pp. 80–89, 2018.
- [18] Z. Mindriawan, “Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kontrol Air Pada Kandang Burung Puyuh Petelur dengan Menggunakan Protokol MQTT,” Universitas Mataram, 2018.
- [19] R. Fitriyanti, A. Aryanti, and Lindawati, “Studi Literatur Mean Opinion Score Menggunakan Moving Picture Quality Metrics (MPQM) Di Jaringan LTE,” *Semin. Nas. Inov. dan Apl. Teknol. di Ind. 2018 Tema A - Penelit.*, vol. 4, no. 2, pp. 10–14, 2018.