

# Sistem *Monitoring* dan *Controlling* Suhu Ruangan Berbasis *Internet of Things*

Fenny Vinola, Abdul Rakhman, Sarjana

Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya  
Jl. Srijaya Negara, Bukit Besar, Ilir Barat 1, Kota Palembang, Sumatera Selatan, 30139, Indonesia

E-mail: fennyvn11@gmail.com, arahmanhamid.60@gmail.com, anna.sarjana@gmail.com

Diterima: 1 Agustus 2020; direvisi: 29 Agustus 2020; disetujui: 31 Agustus 2020

**Abstract - Air Conditioner (AC) is an electronic device that can regulate the room temperature. Unsuitable temperature conditions in a room can cause damage to equipment or items in the room. Remote temperature monitoring and air conditioner control systems can be a solution for monitoring the state of the room.**

**This monitoring and controlling system was built using Internet of Things (IoT) technology. This system uses a Raspberry Pi, DHT22 sensor as a temperature sensor, Infrared as an AC control sensor, and a companion application. In realizing this system, a communication protocol is needed to implement IoT, namely Message Queuing Telemetry Transport (MQTT) protocol.**

**Data from sensors are stored in a database and then the data can be processed and accessed through applications on electronic devices such as Android, iOS, or web-based. By this monitoring and controlling system, it can make easier someone to supervise the air conditioner in a room.**

**Keywords – Controlling; Internet of Things; Monitoring; Raspberry Pi; Temperature Sensor**

**Abstrak – Air Conditioner (AC) merupakan perangkat elektronik yang dapat mengatur suhu suatu ruangan. Kondisi suhu suatu ruangan yang tidak sesuai dapat menimbulkan kerusakan pada peralatan atau barang yang berada pada ruangan tersebut. Sistem monitoring suhu dan pengendali pendingin ruangan jarak jauh dapat menjadi salah satu solusi untuk mengawasi keadaan suatu ruangan.**

**Sistem pemantauan dan pengendalian ini dibangun dengan memanfaatkan teknologi *Internet of Things* (IoT). Sistem ini menggunakan Raspberry Pi, sensor DHT22 sebagai sensor suhu, Infrared sebagai sensor pengendalian AC, dan aplikasi pendamping. Dalam merealisasikan sistem ini dibutuhkan sebuah protokol komunikasi untuk mengimplementasikan IoT yaitu protokol *Message Queuing Telemetry Transport* (MQTT).**

**Data dari sensor disimpan pada database kemudian data tersebut dapat diolah dan diakses melalui aplikasi pada perangkat elektronik seperti Android, iOS, atau berbasis web. Dengan adanya sistem pemantauan dan pengendalian ini, dapat memudahkan seseorang dalam melakukan pengawasan terhadap AC pada suatu ruangan.**

**Kata kunci – Internet of Things; Pemantauan; Pengendali; Raspberry Pi; Sensor Suhu**

## I. PENDAHULUAN

Suhu ruangan umumnya dapat diatur dengan menggunakan alat pendingin ruangan seperti *Air Conditioner* (AC). Kondisi

suhu ruangan yang tidak sesuai khususnya untuk penyimpanan barang atau peralatan dapat mempercepat rusaknya peralatan atau barang yang terdapat pada ruangan tersebut. Oleh karena itu perlu adanya sistem pemantau suhu ruangan dan pengendali alat pendingin ruangan yang praktis, efisien dan dapat dimonitor dari jarak jauh.

Solusi yang dapat digunakan untuk menangani masalah di atas salah satunya adalah dengan memanfaatkan teknologi *Internet of Things* (IoT). Dengan menggunakan IoT, sebuah sensor dapat melakukan pengambilan data dari suatu tempat dan dapat dilakukan akses jarak jauh untuk mengendalikan benda lain di suatu tempat. Hal ini memungkinkan untuk memonitoring suhu ruangan dan mengendalikan alat pendingin ruangan dari jarak jauh melalui perangkat elektronik. Agar informasi dari sensor mengenai suhu ruangan dapat dikirim ke perangkat elektronik maka dibutuhkan sebuah protokol komunikasi.

Salah satu protokol komunikasi yang tepat untuk mengimplementasikan IoT di dalam sistem monitoring jarak jauh yaitu *Message Queuing Telemetry Transport* (MQTT). MQTT merupakan protokol konektivitas *machine-to-machine* (M2M) yang sangat ideal untuk perangkat yang terhubung dengan aplikasi *mobile*. MQTT menggunakan prinsip *publish/subscribe*. Dengan teknologi ini, pemantauan suhu ruangan dan pengendalian AC menjadi lebih praktis dan bisa dilakukan dimana saja.

### A. Sistem Monitoring

Monitoring adalah suatu aktivitas yang dilakukan untuk mengetahui proses jalannya suatu program yang telah dirancang, apakah berjalan dengan baik sesuai dengan yang direncanakan, mengetahui hambatan yang terjadi dan bagaimana cara mengatasi hambatan tersebut [1]. Monitoring bertujuan untuk memastikan apakah suatu proses yang dilakukan sesuai dengan prosedur yang berlaku. Sistem monitoring akan mempermudah suatu pekerjaan jika dirancang dan dilakukan secara efektif. Dalam sistem ini yang dimonitoring adalah suhu ruangan dan aktifitas AC secara *real time*.

### B. Internet of Things

*Internet of Things* (IoT) adalah suatu rancangan yang bertujuan agar perangkat elektronik dapat saling berkomunikasi

secara mandiri, dan dapat menerima serta mengirimkan data menggunakan koneksi jaringan [2]. IoT dapat dimanfaatkan untuk melakukan pemantau dan pengendalian pada suatu tempat tertentu.

Teknologi ini memudahkan orang untuk berbagi hal – hal dengan terkoneksi melalui jaringan baik lokal maupun internet. Beberapa hal yang dapat dilakukan dengan IoT antara lain kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan manusia untuk interaksi manusia atau manusia ke komputer, kemampuan *remote control*, dan sebagainya. Prinsip utama IoT yaitu sebagai sarana untuk memudahkan dalam mengawasi dan mengendalikan sesuatu dengan begitu konsep IoT sangat memungkinkan untuk dapat diterapkan pada kegiatan sehari-hari [3].

### C. Message Queuing Telemetry Transport

*Message Queuing Telemetry Transport* (MQTT) merupakan protokol yang menggunakan prinsip *publish/subscribe* dimana protokol pesannya sangat ringan (*lightweight*) dan *low latency*. MQTT meminimalkan jumlah bytes sebuah jalur dan penggunaan daya yang rendah. Ukuran pesan maksimal 256 MB, namun tidak benar-benar dirancang untuk mengirim sejumlah data yang besar [4].

Jumlah penggunaan energy pada protokol MQTT lebih sedikit dan protokol ini dapat bekerja dengan kondisi *bandwidth* kecil serta *latency* tinggi menjadi alasan utama dalam pemilihan protokol MQTT sebagai protokol untuk mengimplementasikan IoT pada sistem ini [5]. Pada penerapannya, perangkat akan terhubung pada sebuah broker dan mempunyai *topic* tertentu. Pada MQTT, broker berfungsi untuk memproses data *publish* dan *subscribe* dari berbagai perangkat. Broker yang digunakan adalah Mosquitto.

### D. Raspberry Pi

Raspberry Pi merupakan sebuah mini komputer yang dapat digunakan seperti sebuah *Personal Computer* (PC). Raspberry Pi menggunakan *Operating System* (OS) agar bisa dioperasikan. Raspberry Pi menggunakan sistem operasi utama Debian GNU/Linux dan menggunakan bahasa pemrograman Python. Raspberry Pi menggunakan *System On Chip* (SoC) ARM yang dikemas dan diintegrasikan di atas PCB dilengkapi dengan semua fungsi seperti sebuah komputer lengkap. Raspberry Pi terdiri dari Raspberry Pi model A dan model B. Secara umum Raspberry Pi Model B memiliki memori sebesar 512 MB [6]. Pada penelitian ini model Raspberry Pi yang digunakan yaitu Raspberry Pi 3 B+. Raspberry Pi 3 B+ terdiri dari 40 pin GPIO digital port dan USB untuk mengkoneksikan berbagai perangkat USB seperti keyboard, mouse, dan lain-lain. Raspberry Pi 3 model B+ sudah dilengkapi dengan *wifi built-in*.

### E. Hardware (Perangkat Keras)

Selain menggunakan Raspberry Pi, perangkat keras yang digunakan yaitu NodeMCU ESP8266, perangkat ini didasarkan pada modul ESP8266 yang sudah berisikan GPIO, *Pulse Width Modulation* (PWM), dan *Analog to Digital Converter* (ADC) semua dalam satu *board*. NodeMCU adalah sebuah *firmware* yang bersifat *Open-source* dan pengembangan kit yang

membantu untuk membuat prototype produk IOT (*Internet of Things*). NodeMCU menggunakan bahasa *scripting* Lua [7]. NodeMCU ESP8266 dilengkapi berbagai fitur layaknya mikrokontroler dengan kapabilitas akses terhadap WiFi juga chip komunikasi USB to Serial sehingga bisa melakukan programming langsung ke ESP8266 tanpa membutuhkan mikrokontroler tambahan. DHT22 digunakan sebagai sensor untuk membaca suhu dan kelembaban udara. Sensor DHT22 merupakan sensor suhu dan kelembaban relatif dengan keluaran berupa sinyal digital [8]. Selanjutnya perangkat yang digunakan yaitu ESP32 Wemos LoLin32 Lite merupakan modul development board berbasis ESP32. Sensor yang digunakan yaitu *Infrared* LED untuk mengirimkan sinyal perintah ke AC dan *Infrared Receiver* yang bertindak sebagai media untuk menerima sinyal dari *remote* AC.

### F. Software (Perangkat Lunak)

Adapun perangkat lunak yang digunakan yaitu OpenHAB. OpenHAB adalah perangkat lunak *open-source* yang berfungsi sebagai hub yang menyatukan beragam perangkat melalui penggunaan binding [9]. OpenHAB mendukung sejumlah *backend* persistensi untuk menyimpan data. Agar dapat memberikan akses jarak jauh yang aman terdapat OpenHAB *Cloud* yang memungkinkan untuk memonitoring dan mengontrol perangkat-perangkat dari jarak jauh melalui internet.

### G. Penelitian Terkait

Penelitian oleh Rahmi Khalidah dan Nanta Fakhri Pebrianto. 2020. Sistem Pemantauan dan Pengendali Pendingin Ruangan Cerdas Berbasis Cloud dengan Raspberry Pi : Penelitian ini berupa sistem monitoring suhu dan kelembaban ruangan serta controlling suhu pendingin ruangan ditampilkan pada aplikasi *smartphone* dan dapat dikendalikan serta dapat dilakukan jarak jauh [8].

Penelitian oleh Irsandi Satria Wicaksana, Firdaus Iman Ubaidillah, Yeni Prasetyo Hadi, Sandi Tyas Wahyu, Istiadi. 2018. Perancangan Sistem Monitoring Suhu Gudang Berbasis Internet of Things (IoT) : Penelitian ini memberikan gambaran model pemantau pengendali suhu ruang berbasis IoT yang *realtime* dan dimanapun dapat diketahui sehingga kondisi suhu serta aktuatoarnya dapat diketahui setiap saat [10].

Penelitian oleh Periyaldi, Arief Bramanto W.P, Agusma Wajiansyah. 2017. Implementasi Sistem Monitoring Suhu Ruang Server Satnetcom Berbasis *Internet of Things* (IoT) Menggunakan Protokol Komunikasi *Message Queue Telemetry Transport* (MQTT) : Penelitian ini berupa sistem yang mampu mengetahui perubahan suhu ruangan secara *realtime* dan dapat dimonitoring dari jarak jauh. Penelitian ini memberikan kemudahan kepada SatNetCom sebagai perusahaan di bidang penyedia jasa internet (ISP) dalam memonitoring apabila terjadi kenaikan suhu yang signifikan yang dapat mempengaruhi kinerja sistem atau menimbulkan kerusakan pada sisi perangkat [11].

Penelitian oleh Aldy Mangeber, Glanny Mangindaan, Pinrolinvic Manembu, Reynold F. Robot. 2020. *Monitoring* Temperatur Air dan Kecepatan Fan *Cooling water* Pembangkit Listrik Tenaga Panas Bumi Siklus Biner. Penelitian ini

merupakan sistem *monitoring* yang bersifat jarak jauh bertujuan untuk memonitoring temperatur dan kecepatan putar *cooling water* secara *realtime* yang dapat ditampilkan di *website*. [12]

Penelitian oleh Bea Y. Octavian, Glanny M. Ch. Mangindaan, Pinrolinvic D.K Manemby, Reynold F. Robot. 2020. Sistem *Monitoring* Intensitas Cahaya dan Kecepatan Angin Berbasis Web di Kawasan Relokasi Pandu. Dalam penelitian ini, penulis membahas tentang sistem yang dapat memonitoring intensitas cahaya dan kecepatan angin serta parameter keluaran panel surya berbasis web. [13].

## II. METODE

### A. Kerangka Penelitian

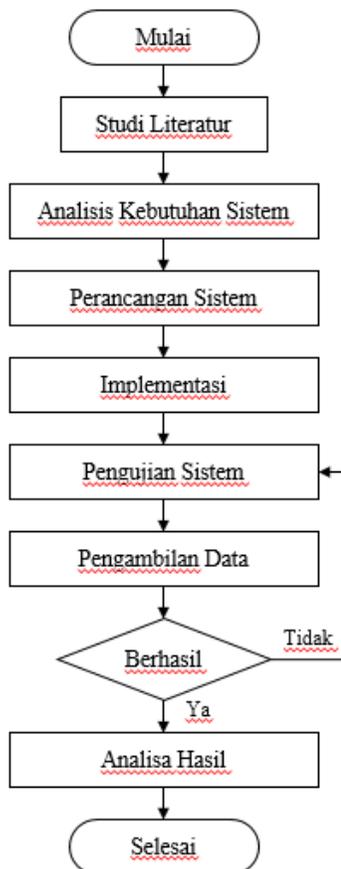
Dalam penelitian ini, untuk merancang sistem *monitoring* dan *controlling* berbasis IoT, tahapan penelitian yang dilakukan dapat dilihat pada gambar 1.

#### 1) Studi Literatur

Pada tahap ini melakukan pencarian, survei, dan membandingkan beberapa referensi yang didapat dari buku, jurnal, maupun riset untuk mendapatkan materi yang sesuai dengan penelitian.

#### 2) Analisis Kebutuhan Sistem

Tahap ini merupakan langkah analisa terhadap apa yang dibutuhkan dalam penelitian. Proses dimana secara sistematis menguraikan pokok-pokok permasalahan yang dihadapi.



Gambar 1. Kerangka Penelitian

#### 3) Perancangan Sistem

Pada tahap ini, melakukan perancangan sistem, mulai dari merancang blok diagram dan skema node (mikrokontroler, sensor) yang akan dibuat.

#### 4) Implementasi

##### a. Server

Melakukan konfigurasi pada Raspberry Pi, mulai dari pemasangan sistem operasi, instalasi *software* aplikasi yang dibutuhkan seperti balenaEtcher, instalasi dan konfigurasi aplikasi untuk memantau dan mengendalikan sensor dari jarak jauh yaitu *framework* OpenHAB.

##### b. Node

Melakukan penulisan algoritma kode untuk selanjutnya diupload ke setiap mikrokontroler.

##### c. User

Mempersiapkan perangkat pemonitoring yaitu OpenHAB dan konfigurasi pada aplikasi OpenHAB di *smartphone*.

#### 5) Pengujian Sistem

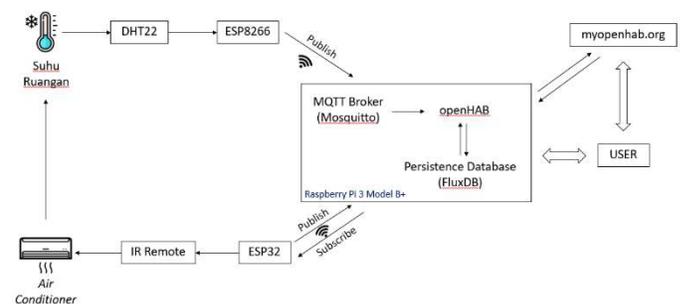
Melakukan pengujian terhadap sensor dan sistem yang bertujuan untuk mengetahui apakah sudah berjalan sesuai dengan yang diharapkan.

#### 6) Pengambilan Data

Melakukan proses pengambilan data terhadap pengujian yang dilakukan.

#### 7) Analisis Data

Analisa data merupakan tahap akhir dari penelitian. Data pengukuran sensor sudah berhasil diolah dan sistem dapat memberikan informasi.



Gambar 2. Diagram Blok Keseluruhan



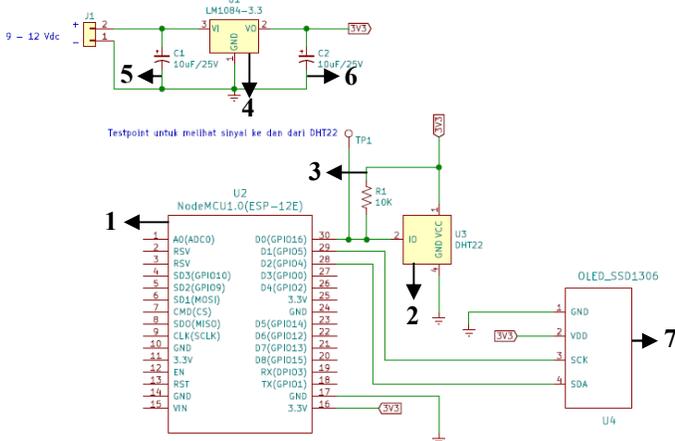
Gambar 3. Box Perangkat Keras

**B. Perancangan Perangkat**

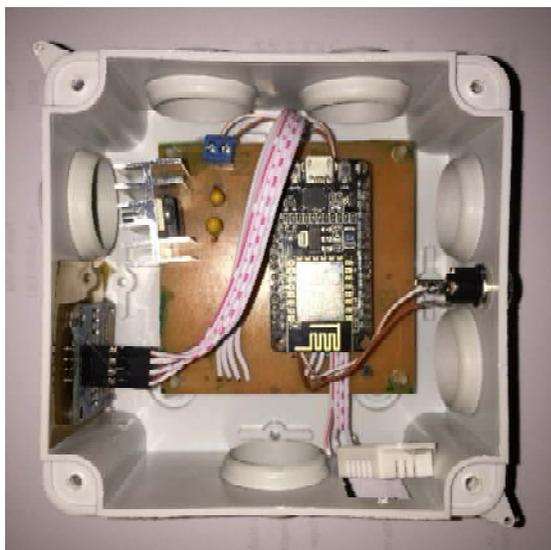
Perancangan perangkat pada penelitian ini dibagi menjadi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat diawali dengan pembuatan diagram blok sistem secara keseluruhan. Alur data dimulai dari pengiriman data berupa pembacaan suhu ruangan oleh sensor DHT22 dan ESP8266 sebagai mikrokontrolernya yang kemudian akan diteruskan ke Raspberry Pi yang berperan sebagai server. Data tersebut akan ditampilkan di sebuah aplikasi pemantauan yang dapat diakses secara lokal maupun publik. Pengguna dapat memberikan perintah kepada perangkat pengendali AC. Perintah tersebut dikirimkan kembali ke Raspberry Pi dan Raspberry Pi akan *unsubscribe* ke ESP32 sebagai mikrokontroler kemudian perintah untuk pengendalian *Air Conditioner* (AC) dikirim melalui *Infrared LED Remote*. Gambaran alur ini dapat dilihat pada gambar 2.

Keterangan :

- 1. NodeMCU ESP8266
- 2. DHT22
- 3. Resistor
- 4. LM1084
- 5. Kapasitor
- 6. Kapasitor
- 7. OLED



Gambar 4. Skematik rangkaian pemantau suhu ruangan



Gambar 5. Hasil rancangan rangkaian pemantau suhu ruangan

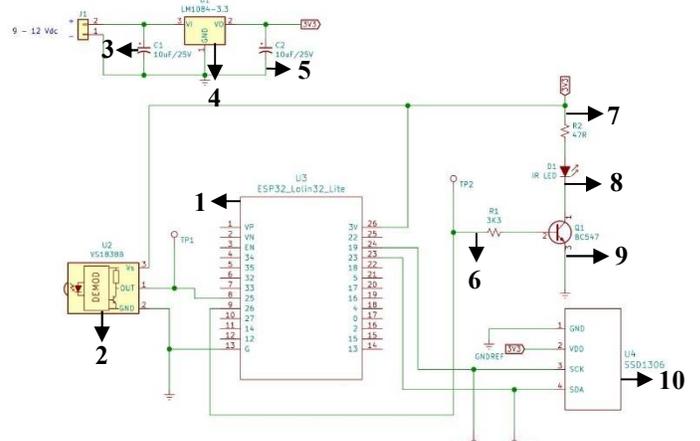
**C. Perangkat Keras**

Untuk perangkat keras terdapat dua perangkat yang dibuat yaitu perangkat pemantau suhu ruangan dan perangkat pengendali AC dapat dilihat pada gambar 3. Pada perangkat pemantau suhu ruangan digunakan DHT22 sebagai sensor untuk membaca suhu pada ruangan dan NodeMCU ESP8266 sebagai mikrokontroler. Rangkaian perangkat pemantau suhu dapat dilihat pada gambar 4 dan gambar 5.

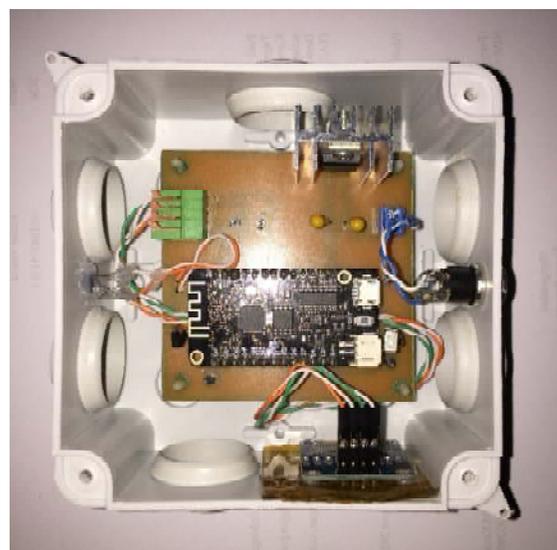
Pada perangkat pengendali AC digunakan *Infrared Receiver Module* yang berfungsi sebagai penerima sinyal dari *remote AC*, *Infrared LED Transmitter* berfungsi sebagai pengirim sinyal yang berupa perintah untuk mengendalikan AC, dan ESP32 Wemos LoLin32 Lite sebagai mikrokontroler seperti yang terlihat pada gambar 6 dan 7.

Keterangan :

- 1. ESP32 LoLin32 Lite
- 2. Infrared Receiver Module
- 3. Kapasitor
- 4. LM1084
- 5. Kapasitor
- 6. Resistor
- 7. Resistor
- 8. Infrared LED
- 9. Transistor BC547
- 10. OLED



Gambar 6. Skematik rangkaian pengendali pendingin ruangan



Gambar 7. Hasil rancangan rangkaian pengendali pendingin ruangan

Untuk dapat menjalankan aplikasi pemantauan (OpenHAB), kita harus mempersiapkan OpenHABian di Raspberry Pi. OpenHABian adalah sistem operasi berbasis Raspbian yang diperuntukan hardware Raspberry Pi dimana di dalamnya sudah disertakan paket-paket dan konfigurasi untuk menjalankan OpenHAB secara mudah.

### 1) Mempersiapkan OpenHABian di Raspberry Pi

Download terlebih dahulu file OpenHABian versi terbaru di website <http://openhbab.org>. file tersebut dalam ekstensi xz (Gambar 8). Kemudian tulis image ke SD Card menggunakan aplikasi balenaEtcher dan lakukan flash (Gambar 9). Setelah OpenHABian otomatis update paket dari repository selesai (Gambar 10), login ke OpenHAB dan kemudian mempersiapkan agar OpenHAB dapat bekerja (Gambar 11).

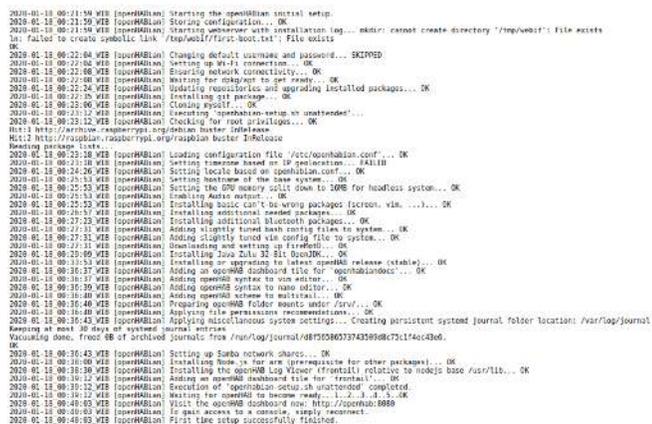


openhbabian-pi-raspbian-201908050414-gitca0976f-crc6a66b5a1.i...

Gambar 8. File OpenHABian



Gambar 9. Tampilan aplikasi balenaEtcher



Gambar 10. OpenHABian automatic update package

### 2) Konfigurasi PuTTY

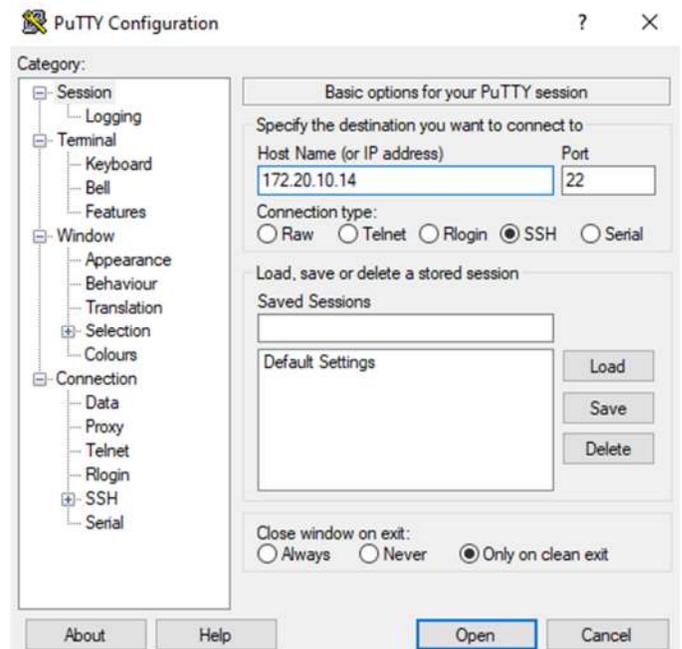
Saat akan menjalankan aplikasi PuTTY, client harus terhubung pada jaringan yang sama dengan Raspberry Pi. Pada Host Name masukkan alamat IP Raspberry Pi. Kemudian pilih SSH seperti gambar 12. Jika alamat IP yang dimasukkan sudah benar dan sudah memilih koneksi SSH maka klik tombol Open. Raspberry Pi sudah siap dioperasikan.

### 3) Pemrograman pada Mikrokontroler

Pada pembuatan program untuk diupload pada mikrokontroler digunakan software Thonny. Bahasa pemrograman yang digunakan adalah bahasa MicroPython. Kode program ditulis dalam file berekstensi .py dan proses compiling dilakukan langsung di dalam mikrokontroler sehingga tidak membutuhkan software tambahan. Program yang telah ditulis akan diupload ke NodeMCU ESP8266 dapat dilihat pada gambar 13 dan diupload ke ESP32 dapat dilihat pada gambar 14.



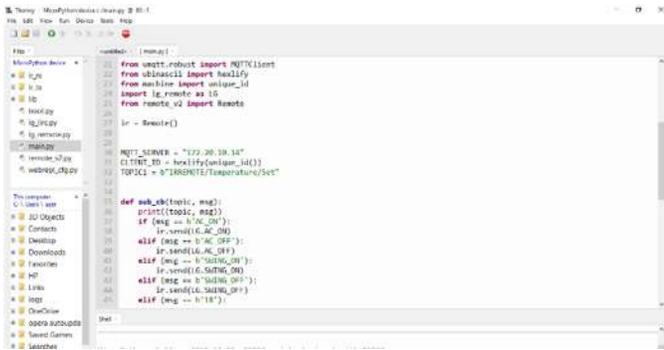
Gambar 11. Tampilan login OpenHAB pada Raspberry Pi



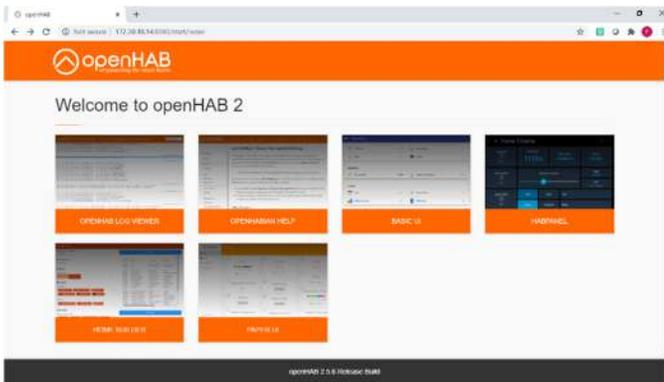
Gambar 12. Konfigurasi PuTTY



Gambar 13. Pemrograman pada NodeMCU ESP8266 untuk Sensor DHT22



Gambar 14. Pemrograman pada ESP32 untuk Sensor *Infrared*



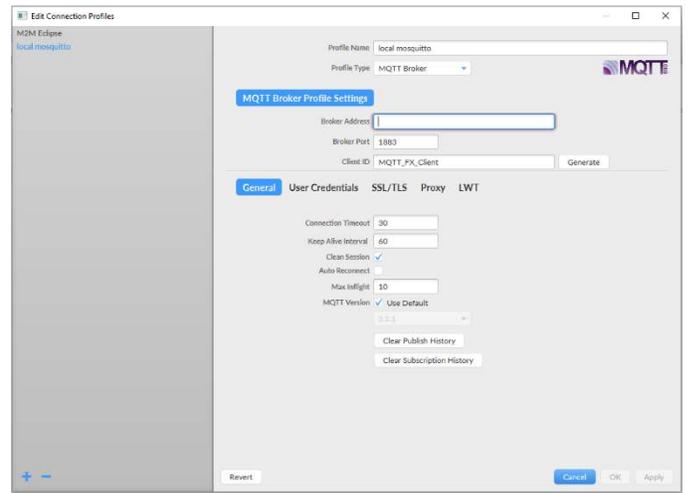
Gambar 15. Tampilan awal user interface OpenHAB

*D. Perangkat Lunak*

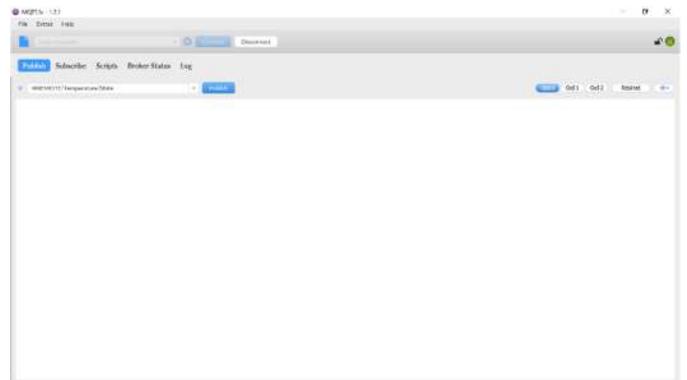
Data-data hasil pembacaan sensor kemudian akan ditampilkan dalam bentuk grafik berupa hasil visualisasi oleh Grafana yang diperoleh dari data yang tersimpan pada InfluxDB *Persistence*. Sebelum itu, untuk dapat mengakses perangkat keras melalui aplikasi pemantauan (OpenHAB) harus melakukan konfigurasi terlebih dahulu pada OpenHAB.

Untuk menambahkan hal-hal baru, harus melakukan instal *Bindings* yang dibutuhkan seperti MQTT Bindings dan NTP Bindings untuk pengaturan waktu, menginstal *Persistence* agar data dapat tersimpan sepanjang waktu pada OpenHAB, menambahkan *Things*, *Channel*, dan *Items* pada *Bindings* sesuai dengan kebutuhan dapat dilakukan pada halaman PAPER UI seperti yang terlihat pada gambar 15.

Selain itu perlu dilakukan uji komunikasi MQTT dan OpenHAB. Pengujian terhadap implementasi protokol MQTT pada OpenHAB dilakukan menggunakan aplikasi MQTT.fx. Masukkan *Broker Address* yang digunakan agar dapat terhubung (Gambar 16). Dengan begitu kita mem-*publish* dan men-



Gambar 16. Tampilan pengaturan aplikasi MQTT.fx



Gambar 17. Tampilan aplikasi MQTT.fx setelah terhubung

*subscribe* informasi untuk dapat mengetahui apakah protokol MQTT sudah terimplementasikan pada OpenHAB (Gambar 17). Masukkan MQTT *State Topic* atau MQTT *Command Topic* yang sudah kita tentukan sebelumnya pada OpenHAB sehingga data yang atau informasi yang di-*publish* dan di-*subscribe* akan terbaca pada aplikasi OpenHAB.

*1) Menghubungkan OpenHAB ke OpenHAB Cloud*

Untuk dapat terhubung dan diakses secara publik, OpenHAB lokal harus dihubungkan ke OpenHAB *Cloud*. Salah satu mesin virtual dari layanan OpenHAB *Cloud* adalah myopenHAB.org. Langkah pertama yang harus dilakukan adalah menginstal OpenHAB *Cloud Connector*. Kemudian buka aplikasi PuTTY, pilih SSH dan masukkan IP *Address* atau *Host Name* kemudian klik *Open* seperti pada gambar 12. Untuk dapat mengautentikasi dengan layanan OpenHAB *Cloud* dibutuhkan kode *UUID* dan *Secret*. *UUID* adalah sebuah pengidentifikasi unik yang mirip seperti nama pengguna untuk otentikasi cloud. Sedangkan *Secret* adalah kunci rahasia acak yang berfungsi sebagai kata sandi. Untuk mendapatkan *UUID* ketik cat `/var/lib/openHAB2/uuid` pada tampilan PuTTY lalu enter, dan untuk mendapatkan *Secret* ketik cat `/var/lib/openHAB2/openHABcloud/secret` lalu enter. *UUID* dan *Secret* akan tampil pada aplikasi PuTTY seperti yang terlihat pada gambar 18. Pada myopenHAB.org, masukkan email, password, OpenHAB *UUID*, dan OpenHAB *Secret*, lalu

klik Register (Gambar 19), kemudian mengkonfigurasi ulang UUID dan *Secret* di bagian *My Account*. Lalu klik *Update*. Dengan begitu OpenHAB lokal sudah terhubung ke OpenHAB *Cloud* (Gambar 20).

### III. HASIL DAN PEMBAHASAN

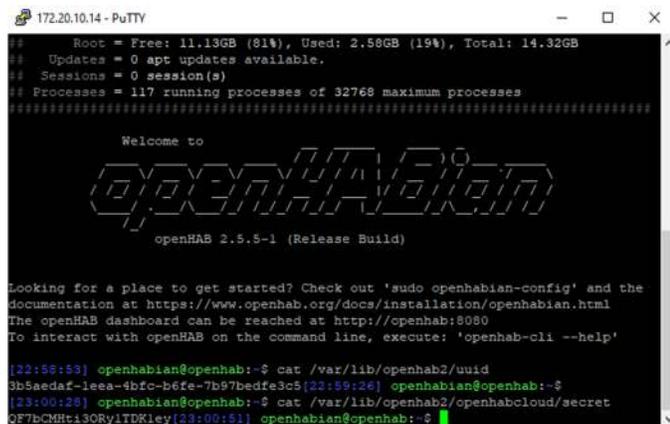
#### A. Hasil Perancangan

##### 1) Monitoring Suhu Ruangan

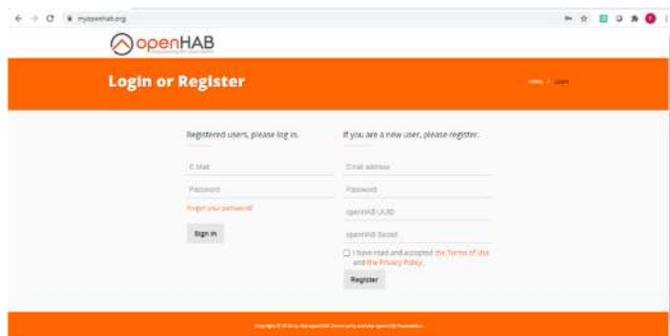
Tampilan akhir dari hasil perancangan sistem *monitoring* suhu ruangan dengan menggunakan aplikasi OpenHAB. Gambar 21 menampilkan periode grafik pembacaan suhu dan kelembaban ruangan untuk pembacaan 5 menit terakhir dan Gambar 22 menampilkan fitur dari periode grafik pembacaan suhu dan kelembaban ruangan untuk pembacaan satu hari terakhir.

##### 2) Controlling Pendingin Ruangan

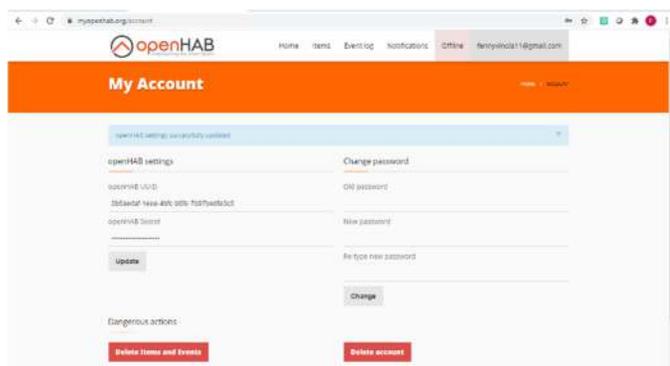
Tampilan akhir dari hasil perancangan sistem *controlling* perangkat pendingin ruangan dengan menggunakan aplikasi OpenHAB dapat dilihat pada gambar 23. Pada tampilan *controlling* menggunakan aplikasi OpenHAB. Terdapat beberapa tombol yang bisa digunakan untuk mengatur jalannya pendingin ruangan atau AC yaitu tombol untuk mengatur hidup atau matinya AC (*Remote Control Power*), tombol untuk mengatur suhu AC yang diinginkan (*Remote Control Temperature Set*), dan tombol untuk mengatur *Swing Mode*. Setelah dilakukan pengujian, dapat diketahui bahwa semua tombol (*button*) dapat memberikan perintah dan berfungsi dengan baik yang dapat dilihat pada Tabel I.



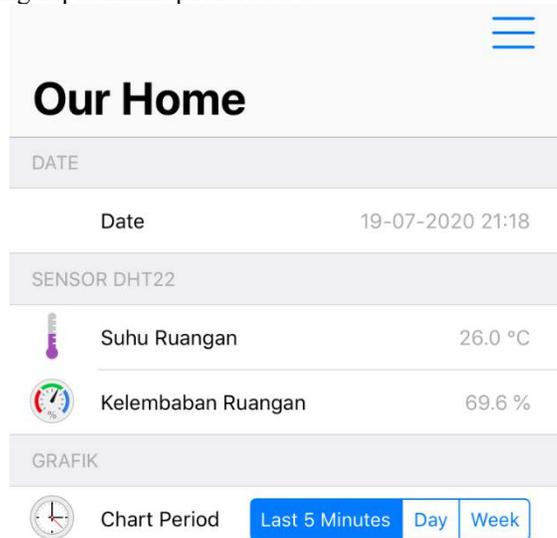
Gambar 18. Kode UUID dan *Secret* OpenHAB *Cloud*



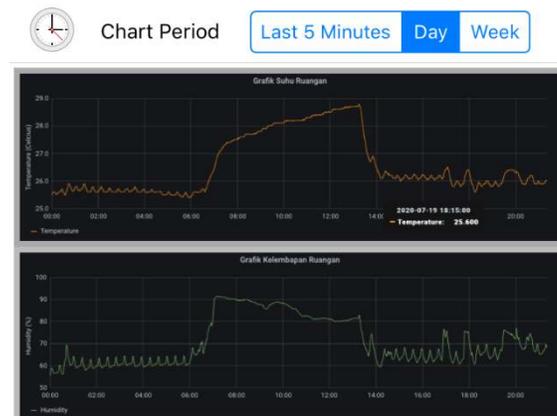
Gambar 19. Tampilan *Register* myopenHAB.org



Gambar 20. Tampilan *My Account* myopenHAB.org



Gambar 21. Tampilan periode grafik untuk pembacaan 5 menit terakhir



Gambar 22. Tampilan periode grafik untuk pembacaan 1 hari terakhir

Gambar 23. Tampilan *controlling* menggunakan aplikasi OpenHABTABEL I  
FUNGSI TOMBOL CONTROL AC

Nama Tombol	Fungsi	Pengujian
Power ON	Menghidupkan AC	Berhasil
Power OFF	Mematikan AC	Berhasil
Remote Control Temp Set	Mengatur suhu AC (Terdapat pengaturan suhu 18°C hingga 27°C)	Berhasil
Swing ON	Menghidupkan mode Swing	Berhasil
Swing OFF	Mematikan mode Swing	Berhasil

### B. Hasil Monitoring Suhu

Penggunaan InfluxDB *Persistence* dan Grafana sebagai penyimpanan data dan memvisualkan data ke dalam grafik yang dapat diakses melalui <http://openhab/3000>. Data yang tersimpan berupa data pembacaan suhu oleh DHT22 yang dikirimkan setiap 10 detik oleh NodeMCU ESP8266 pada sistem monitoring suhu. Semua data hasil pembacaan sensor sejak awal sistem *monitoring* diaktifkan akan tersimpan dan dapat diakses dikemudian hari. Pengambilan data hasil pembacaan sensor dilakukan dua kali yaitu pada pagi dan malam hari. Berikut ini merupakan data hasil pembacaan DHT22 di ruangan berukuran 3 x 9 m<sup>2</sup>.

#### 1) Sistem monitoring suhu pada malam hari

Tabel II menunjukkan bahwa pengambilan data dilakukan selama 2 jam pada tanggal 17 Juli 2020 pukul 00:00 s/d 02:00 WIB. AC diatur dengan suhu 25°C. Data yang terbaca oleh sensor mendekati suhu AC. Hal ini dikarenakan suhu ruangan pada

TABEL II  
HASIL MONITORING SUHU PADA MALAM HARI

No	Pengaturan Suhu AC	Jam (WIB)	Nilai
1	Power ON (25°C) Swing OFF	00:00	25.3 °C
2	Power ON (25°C) Swing OFF	00:10	25.2 °C
3	Power ON (25°C) Swing OFF	00:20	25.2 °C
4	Power ON (25°C) Swing OFF	00:30	25.1 °C
5	Power ON (25°C) Swing OFF	00:40	25.1 °C
6	Power ON (25°C) Swing OFF	00:50	25.1 °C
7	Power ON (25°C) Swing OFF	01:00	25.3 °C
8	Power ON (25°C) Swing OFF	01:10	25.1 °C
9	Power ON (25°C) Swing OFF	01:20	25.2 °C
10	Power ON (25°C) Swing OFF	01:30	25.1 °C
11	Power ON (25°C) Swing OFF	01:40	25.1 °C
12	Power ON (25°C) Swing OFF	01:50	25.1 °C
13	Power ON (25°C) Swing OFF	02:00	25 °C

TABEL III  
HASIL MONITORING SUHU PADA PAGI HARI

No	Pengaturan Suhu AC	Jam (WIB)	Nilai
1	Power ON (20°C) Swing OFF	9:10	26.4 °C
2	Power ON (20°C) Swing OFF	9:20	26.4 °C
3	Power ON (20°C) Swing OFF	9:30	26.25 °C
4	Power ON (20°C) Swing OFF	9:40	25.8 °C
5	Power ON (20°C) Swing OFF	9:50	25 °C
6	Power ON (20°C) Swing OFF	10:00	24.8 °C
7	Power ON (20°C) Swing OFF	10:10	24.4 °C
8	Power ON (20°C) Swing OFF	10:20	24.7 °C
9	Power ON (20°C) Swing OFF	10:30	25.4 °C
10	Power ON (18°C) Swing OFF	10:40	24.6 °C
11	Power ON (18°C) Swing OFF	10:50	24.2 °C
12	Power ON (18°C) Swing OFF	11:00	24 °C
13	Power ON (18°C) Swing OFF	11:10	23.8 °C

TABEL IV  
PENGUJIAN JARAK PANCAR MAKSIMUM INFRA RED

No	Jarak Infrared terhadap AC (cm)	Keterangan
1	30	Sinyal dapat diterima AC
2	60	Sinyal dapat diterima AC
3	90	Sinyal dapat diterima AC
4	120	Sinyal dapat diterima AC
5	150	Sinyal dapat diterima AC
6	180	Sinyal dapat diterima AC
7	210	Sinyal dapat diterima AC
8	240	Sinyal dapat diterima AC
9	270	Sinyal dapat diterima AC
10	290	Sinyal dapat diterima AC
11	320	Sinyal dapat diterima AC
12	350	Sinyal dapat diterima AC
13	380	Sinyal dapat diterima AC
14	390	Sinyal dapat diterima AC
15	400	Sinyal tidak dapat diterima AC

malam hari lebih stabil dibandingkan dengan siang hari karena tidak ada panas dari sinar matahari. Data hasil pembacaan oleh sensor tersimpan pada *persistence* kemudian akan tersimpan sebagai *history* dan dapat diakses kapanpun. Grafik data hasil pembacaan sensor dapat dilihat pada Gambar 24.

#### 2) Sistem monitoring suhu pada pagi hari

Tabel III menunjukkan bahwa pengambilan data dilakukan selama 2 jam pada tanggal 17 Juli 2020 pukul 09:10 s/d 11:00 WIB. AC diatur dengan suhu 20°C pukul 09:10 s/d 10:30 WIB dan diturunkan menjadi 18°C pukul 10:40 s/d 11:10 WIB. Setelah penurunan kondisi suhu AC, terjadi pula penurunan pada pembacaan oleh sensor. Data yang terbaca oleh sensor memiliki nilai suhu yang berbeda jauh dengan suhu yang diatur



Gambar 24. Tampilan grafik suhu ruangan



Gambar 25. Tampilan grafik suhu ruangan



Gambar 26. Diagram waktu data *signal* dari DHT22

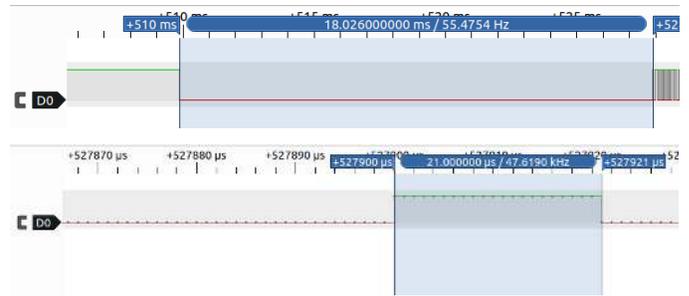
pada pendingin ruangan atau AC. Hal ini dikarenakan adanya sinar matahari yang mempengaruhi suhu ruangan pada pagi hari. Namun perbedaan ini masih dalam batas normal. Data hasil pembacaan tersimpan pada *persistence*. Grafik data hasil pembacaan sensor dapat dilihat pada Gambar 25.

### C. Cara Kerja DHT22

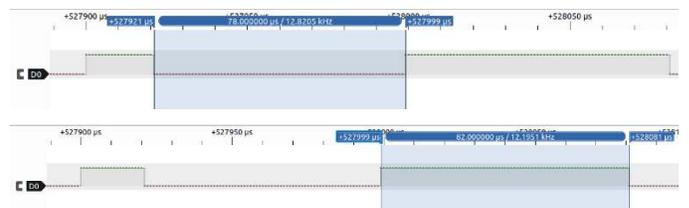
Gambar 26 menunjukkan hasil pembacaan suhu oleh DHT22. Untuk mulai meminta DHT22 membaca data suhu, mikrokontroler harus melakukan *start signal*, yang merupakan logika 0 (rendah) selama minimal 500  $\mu s$  diikuti dengan logika 1 (tinggi) dapat dilihat pada gambar 27. DHT22 kemudian harus merespons (gambar 28) dengan menarik bus rendah selama 80  $\mu s$  lalu menarik bus tinggi selama 80  $\mu s$  lagi sebelum menariknya rendah lagi. DHT22 akan mengirimkan nilai kelembapan terlebih dahulu diikuti dengan nilai suhu. Keduanya adalah data 16-bit. Logika 0 adalah pulsa rendah sepanjang 50  $\mu s$  diikuti pulsa tinggi sepanjang 26  $\mu s$  (gambar 29). Sementara itu, logika 1 adalah pulsa rendah sepanjang 50  $\mu s$  diikuti oleh pulsa tinggi sepanjang 70  $\mu s$  (gambar 30). Berikut ini merupakan hasil pembacaan DHT22 dengan menggunakan *Logic Analyzer*.

Data 40 bit dari DHT22 adalah sebagai berikut :

00000011 00011110 00000001 00011101 00111111



Gambar 27. *Start Signal* dari MCU



Gambar 28. *Response Signal* dari DHT22



Gambar 29. Data Logika 0



Gambar 30. Data Logika 1

Pengubahan 16 bit data *humidity* dari sistem biner ke sistem desimal,

$$0000001100011110 \rightarrow 798$$

$$RH = 798/10 = 79,8\% RH$$

Maka, kelembapan yang terbaca adalah 79,8%

Pengubahan 16 bit data *temperature* dari sistem biner ke sistem desimal,

$$0000000100011101 \rightarrow 285$$

$$T = 285/10 = 28,5 \text{ } ^\circ\text{C}$$

Maka, suhu yang tebaca adalah 28,5 °C

$$\begin{aligned} \text{Parity} &= 00000011 + 00011110 + 00000001 + 00011101 \\ &= 00111111 \end{aligned}$$

#### D. Hasil Pengujian Controlling AC

Dalam proses pengujian perangkat *controlling* AC, pengujian yang dilakukan adalah pengujian terhadap sensor *infrared* dan pengujian *functional suitability* pada aplikasi pengendali AC. Pengujian terhadap sensor *infrared* dilakukan untuk mengetahui jarak pancar maksimal sinyal pada perangkat yang telah dirancang dan dibuat sehingga mampu diterima oleh AC. Hasil pengujian dapat dilihat pada Tabel IV. Berdasarkan Tabel IV dapat diketahui bahwa jarak maksimum *infrared* dari AC agar sinyal yang dipancarkan dapat diterima AC sehingga dapat memberikan perintah kepada AC yaitu pada jarak 3,9 meter. Sedangkan pengujian *functional suitability* dilakukan bertujuan untuk mengetahui kemampuan fitur pengendali AC pada perangkat lunak untuk menyediakan fungsi sesuai dengan kebutuhan pengguna ketika digunakan dalam kondisi tertentu. Hasil pengujian dapat dilihat pada Tabel I. Fitur pengendali AC dapat memberikan perintah dan bekerja sesuai dengan fungsinya jika peletakan perangkat *controlling* AC berada pada jarak yang tepat yaitu tidak lebih dari 3,9 meter ke AC.

## IV. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan hasil perancangan, pembuatan dan pengujian yang telah dilakukan, dapat disimpulkan sistem *monitoring* dan *controlling* dengan menggunakan sensor DHT22 dan *infrared* dapat berjalan dengan baik. *Monitoring* suhu dan *controlling* AC dapat dilakukan melalui aplikasi dengan menggunakan fasilitas IoT. Seluruh data hasil pembacaan sensor dapat tersimpan pada *persistence*. Jarak maksimal *infrared* dan AC agar dapat mengirim dan menerima sinyal yaitu pada jarak 2,9 meter.

### B. Saran

Dalam penelitian ini tentunya masih memiliki kekurangan sehingga diharapkan dapat dilakukan pengembangan dan penyempurnaan seperti pada sistem *controlling* AC. Sebaiknya ditambahkan komponen berupa kamera sehingga keadaan AC dapat termonitor dengan jelas.

## V. KUTIPAN

- [1] R. A. Usman, H. Bambang, and Y. M. Maulana, "Analisis Dan Desain Sistem Monitoring Dan Evaluasi Koperasi Pada Dinas Koperasi Kabupaten Sidoarjo," *JSIKA*, vol. 5, no. 6, pp. 1–8, 2016.
- [2] 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.

- [3] M. I. Kamil, R. A. P. I. Prasetya, and D. Wibawa, "Prototipe Sistem Monitoring Dan Kontrol Lampu Rumah Berbasis Iot ( Internet of Things ) Prototype of Iot-Based Home Light Monitoring and Control Systems," in *e-Proceeding of Engineering*, 2019, vol. 6, no. 2, pp. 2974–2981.
- [4] M. S. Daulah, D. Syauqy, and R. Primananda, "Implementasi Protokol MQTT Pada Monitoring Suhu Dan Ketersediaan Pakan Ikan Pada Akuarium," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2661–2668, 2018.
- [5] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 7521–7527, 2018.
- [6] F. Sirait, "Sistem Monitoring Keamanan Gedung Berbasis Raspberry Pi," *J. Teknol. Elektro*, vol. 6, no. 1, pp. 55–60, 2015, doi: 10.22441/jte.v6i1.790.
- [7] D. P. A. R. Hakim, A. Budijanto, and B. Widjanarko, "Sistem Monitoring Penggunaan Air PDAM pada Rumah Tangga Menggunakan Mikrokontroler NODEMCU Berbasis Smartphone ANDROID," *J. IPTEK, Univ. Widya Kartika*, vol. 22, no. 2, pp. 9–18, 2018, doi: 10.31284/j.ipitek.2018.v22i2.
- [8] R. Khalidah and N. F. Prebianto, "Sistem Pemantauan dan Pengendali Pendingin Ruangan Cerdas Berbasis Cloud dengan Raspberry Pi," *J. Appl. Electr. Eng.*, vol. 4, no. 1, pp. 16–19, 2020.
- [9] J. A. S. Velázquez, "Securing openHAB Smart Home through User Authentication and Authorization," UNIVERSITY OF TARTU, 2018.
- [10] I. S. Wicaksana, F. I. Ubaidillah, Y. P. Hadi, S. T. Wahyu, and Istiadi, "Perancangan Sistem Monitoring Suhu Gudang Berbasis Internet of Things (Iot)," *Conf. Innov. Appl. Sci. Technol. (CIASTECH 2018)*, no. September, pp. 503–511, 2018.
- [11] P. Periyaldi, A. Bramanto, and A. Wajiansyah, "Implementasi Sistem Monitoring Suhu Ruang Server Satnetcom Berbasis Internet of Things (Iot) Menggunakan Protokol Komunikasi Message Queue Telemetry Transport (Mqtt)," *JTT (Jurnal Teknol. Terpadu)*, vol. 6, no. 1, p. 23, 2018, doi: 10.32487/jtt.v6i1.435.
- [12] A. Mangeber, G. Mangindaan, P. Manembu, and R. F. Robot, "Monitoring Temperatur Air dan Kecepatan Fan Cooling water Pembangkit Listrik Tenaga Panas Bumi Siklus Biner," *J. Tek. Elektro dan Komput.*, vol. 9, no. 1, pp. 1–10, 2020, doi: 10.35793/jtek.9.1.2020.28583.
- [13] B. Y. Octavian, P. D. K. Manembu, R. F. Robot, T. Elektro, and U. Sam, "Sistem Monitoring Intensitas Cahaya dan Kecepatan Angin Berbasis Web di Kawasan Relokasi Pandu," *J. Tek. Elektro dan Komput.*, vol. 9, no. 1, pp. 21–30, 2020, doi: 10.35793/jtek.9.1.2020.28571.

## TENTANG PENULIS



Penulis bernama lengkap Fenny Vinola. Anak pertama dari dua bersaudara. Lahir di Lais, 11 September 1998. Alamat tempat tinggal penulis di Jl. HM. Saleh Nomor 88 RT 40 RW 09 Sukarami, Kota Palembang. Penulis menempuh pendidikan formal di SD Muhammadiyah Teluk (2004-2009), SD Negeri 1 Lais (2009-2010), SMP Negeri 6 Unggul Sekayu (2010-2013), SMA Negeri 2 Sekayu (2013-2016). Pada tahun 2016, penulis melanjutkan Pendidikan D4 di Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya. Dalam menempuh pendidikan ini penulis melaksanakan Kerja Praktek selama 2 bulan di Perum LPPNPI Kantor Cabang Palembang dan 6 bulan di PT. Surveyor Indonesia.