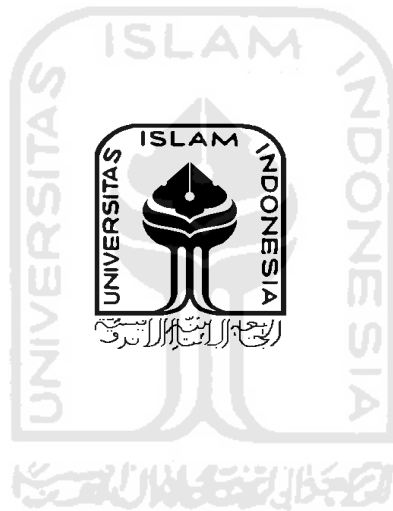


**PEMANFAATAN INTERNET OF THINGS SEBAGAI
KENDALI OTOMATIS PENGKONDISI UDARA RUANG
KELAS**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun Oleh :

Nama : Fauzi Hidayat Putra Mukti

No. Mahasiswa : 16525111

NIRM : 2016080701

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2020

LEMBAR PENGESAHAN DOSEN PEMBIMBING

PEMANFAATAN INTERNET OF THINGS SEBAGAI
KENDALI OTOMATIS PENGKONDISI UDARA RUANG
KELAS

TUGAS AKHIR



Disusun Oleh :

Nama : Fauzi Hidayat Putra Mukti
No. Mahasiswa : 16525111
NIRM : 2016080701

Yogyakarta, 16 September 2020

Pembimbing,

A handwritten signature in black ink, appearing to read 'Agung Nugroho'.

Agung Nugroho Adi S.T., M.T.

LEMBAR PENGESAHAN DOSEN PENGUJI

**PEMANFAATAN INTERNET OF THINGS SEBAGAI
KENDALI OTOMATIS PENGKONDISI UDARA RUANG
KELAS**

TUGAS AKHIR

Disusun Oleh :

Nama : Fauzi Hidayat Putra Mukti

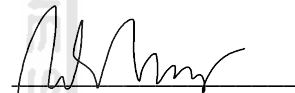
No. Mahasiswa : 16525111

NIRM : 2016080701

Tim Penguji

Agung Nugroho Adi S.T., M.T

Ketua



Tanggal : 14/10/20

Purtojo S.T, M.Sc.

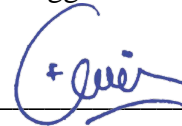
Anggota I



Tanggal : 15/10/202

Mohammad Faizun S.T.,M.Eng.,Ph.D

Anggota II



Tanggal : 14/10/2020

Mengetahui

Ketua Jurusan Teknik Mesin



Dr. Eng. Risdiono, S.T., M.Eng.

HALAMAN PERSEMBAHAN

Tugas akhir ini saya persembahkan untuk Ibu saya, yang selalu mendoakan, menyiapkan seluruh kebutuhan, sandang pangan dan papan, hingga membiayai seluruh pendidikan.

Kepada kakak perempuan saya yang membantu dan mau direpotkan ketika saya memberi pertanyaan dan perlu jawaban.

Kepada dosen pembimbing saya bapak Agung Nugroho Adi S.T., M.T. yang telah membantu dan memberi motivasi dalam mengerjakan.

Kepada teman-teman yang terkadang memberi semangat dan terkadang menjatuhkan, yang menerima saya ketika saya mampir kekontrakan walupun hanya menambah kegabutan.

Kepada seseorang disana yang memberi motivasi tak terlihat dan mungkin dikala sujudnya untuk kesuksesan saya selalu mendoakan.



HALAMAN MOTTO

• بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

1. Dengan nama Allah Yang Maha Pengasih, Maha Penyayang

• الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

2. Segala puji bagi Allah, Tuhan seluruh alam,

• الرَّحْمَنِ الرَّحِيمِ

3. Yang Maha Pengasih, Maha Penyayang,

• مَلِكِ يَوْمِ الدِّينِ

4. Pemilik hari pembalasan.

• إِلَيْكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ

5. Hanya kepada Engkau kami menyembah dan hanya kepada Engkau kami mohon pertolongan.

• اِهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ

6. Tunjukilah kami jalan yang lurus

• صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ ۗ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ

7. (yaitu) jalan orang-orang yang telah Engkau beri nikmat kepadanya; bukan (jalan) mereka yang dimurkai, dan bukan (pula jalan) mereka yang sesat.

KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Allah SWT, yang telah memberikan rahmat serta karunianya kepada penulis hingga mampu menyelesaikan laporan tugas akhir sebagai salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi Mesin Fakultas Teknologi Industri Universitas Islam Indonesia.

Sholawat serta salam selalu diberikan kepada Nabi besar kita Muhammad SAW yang telah membimbing umat dari zaman kegelapan menuju zaman yang sesuai dengan ketentuan Allah SWT. Serta selama pelaksanaan tugas akhir, penyusun ingin berterima kasih kepada:

1. Orang tua tercinta serta Saudari penulis yang selalu mendo'akan dan membantu segala sesuatunya.
2. Bapak Agung Nugroho Adi S.T., M.T. selaku Dosen Pembimbing tugas akhir yang telah mendampingi dan memberikan bantuan pikiran dan materi.
3. Dan Jurusan Teknik Mesin Universitas Islam Indonesia, yang telah memfasilitasi dan mediasi proses pembuatan tugas akhir

Saya menyadari sepenuhnya bahwa dalam mengerjakan tugas akhir ini banyak kekeliruan dan kekurangan, untuk itu saya mohon maaf sebesar-besarnya semoga hasil dari tugas akhir ini dapat bermanfaat bagi semua pihak yang membutuhkan di masa yang akan datang, Amin.

Penulis

Yogyakarta, 16 September 2020

ABSTRAK

AC (Air Conditioner) sekarang menjadi kebutuhan penting dalam menunjang kenyamanan perkuliahan disuatu ruang kelas. Settingan suhu AC dapat diubah-ubah sesuai kebutuhan untuk kenyamanan penggunaan suatu ruangan. Namun penggunaan AC yang tidak terkontrol, seperti kejadian ketika ruangan sudah tidak digunakan namun AC masih menyala, mengakibatkan pemborosan baik uang maupun energi. Sehingga harus dibuat sistem atau alat yang mampu mengontrol AC secara otomatis agar mengurangi pembrosan. Konsep *Internet of Things (IoT)* yang mampu menghubungkan seluruh alat elektronik ke internet dirasa mampu untuk diterapkan. Prototipe alat direalisasikan dengan menggunakan mikrokontroler ESP32 yang sudah dilengkapi dengan *wifi* dan *Bluetooth* yang digunakan sebagai otak dan aktuator dari alat kontrol AC ruangan. Alat menggunakan cahaya inframerah sebagai pemberi sinyal dari ESP32 ke AC. Ditambahkan dengan sensor input berupa sensor PIR untuk pendeteksi gerakan dan juga sensor suhu DHT11 sebagai sensor monitoring suhu ruangan. Penggunaan protokol MQTT dengan broker Mosquitto yang berbasis *publish* dan *subscribe* membuat komunikasi antara sensor, mikrokontroler dan NodeRED sebagai webserver sangat mudah. Alat dirancang agar mampu dikontrol secara manual melalui tampilan antar muka dari NodeRED dan juga secara otomatis. Dari ujicoba seluruh sistem, alat mampu menghidupkan dan mematikan dua buah AC yang ada diruangan menggunakan IR Led dengan jarak kurang lebih 4.5 m dari alat, sekaligus mensetting suhu AC secara otomatis. Sensor PIR bekerja sesuai yang diharapkan dengan jarak kemampuan kurang lebih 2.5 meter. Tampilan antar muka NodeRED mampu menampilkan kontrol AC secara manual dan juga menampilkan monitoring suhu ruangan secara *real time*. Tampilan antar muka NodeRED bisa dilihat dan digunakan pada laptop/PC dan smartphone dengan memasukkan *IP address* NodeRED.

Kata kunci: IoT, Pengkondisi Udara, NodeRED, ESP32, MQTT

ABSTRACT

AC (Air Conditioner) is now an important requirement in supporting the comfort of lectures in a classroom. AC temperature settings can be changed according to the needs for comfortable use of a room. However, uncontrolled use of AC, such as the incident when the room is no longer in use but the AC is still on, results in a waste of both money and energy. So it must be made a system or device that is able to control the air conditioner automatically in order to reduce bulking. The concept of the Internet of Things (IoT), which is able to connect all electronic devices to the internet, is considered capable of being applied. The prototype of the device is realized using an ESP32 microcontroller which is equipped with wifi and Bluetooth which is used as the brain and actuator of the room air conditioner controller. The tool uses infrared light as a signal from the ESP32 to the AC. Added with an input sensor in the form of a PIR sensor for motion detection and also a DHT11 temperature sensor as a room temperature monitoring sensor. The use of the MQTT protocol with the Mosquitto broker based on publish and subscribe makes communication between sensors, microcontrollers and NodeRED as a webserver very easy. The tool is designed to be able to be controlled manually via the NodeRED interface as well as automatically. From testing the entire system, the tool is able to turn on and turn off two air conditioners in the room using an IR LED with a distance of approximately 4.5 m from the appliance, as well as setting the AC temperature automatically. The PIR sensor works as expected with a distance of approximately 2.5 meters. The NodeRED interface is capable of displaying manual AC controls and also displays real time room temperature monitoring. The NodeRED interface can be viewed and used on laptops / PCs and smartphones by entering the NodeRED IP address.

Keywords: IoT, Air Conditioner, NodeRED, ESP32, MQTT

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan Dosen Pembimbing	ii
Lembar Pengesahan Dosen Penguji	iii
Halaman Persembahan	iv
Halaman Motto	v
Kata Pengantar.....	vi
Abstrak	vii
Abstract.....	viii
Daftar Isi	ix
Daftar Tabel.....	xii
Daftar Gambar	xiii
Daftar Notasi.....	xvi
Bab 1 Pendahuluan	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian atau Perancangan	3
1.5 Manfaat Penelitian atau Perancangan	3
1.5.1 Manfaat Penelitian bagi Mahasiswa.....	3
1.5.2 Manfaat Penelitian bagi Universitas.....	3
1.6 Sistematika Penulisan	4
1.6.1 BAB I – Pendahuluan.....	4
1.6.2 BAB II – Tinjauan Pustaka.....	4
1.6.3 BAB III – Metode Penelitian.....	4
1.6.4 BAB IV – Hasil dan Pembahasan.....	4
1.6.5 BAB V – Penutup	4
Bab 2 Tinjauan Pustaka	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori	7
2.2.1 NodeRed.....	7

2.2.2	MQTT	8
2.2.3	ESP32	10
2.2.4	PIR Sensor	11
2.2.5	IR LED (Infra Red LED).....	12
Bab 3 Metode Penelitian		13
3.1	Alur Penelitian	13
3.2	Peralatan dan Bahan.....	14
3.2.1	Perangkat Keras	14
3.2.2	Perangkat Lunak	14
3.2.3	Bahan	14
3.3	Perancangan Sistem Remote.....	15
3.4	Perancangan sistem Publish dan Subscribe MQTT	19
3.5	Perancangan Program ESP32	20
3.6	Perancangan Flow Node pada NodeRED	25
3.7	Perancangan rangkaian perangkat keras	26
3.8	Desain 3D Alat.....	28
Bab 4 Hasil dan Pembahasan		29
4.1	Pembahasan	29
4.1.1	Pembahasan Program ESP32	29
4.1.2	Command Prompt Windows	36
4.2	Hasil Pengujian	37
4.2.1	Hasil Pengujian Sensor PIR.....	37
4.2.2	Hasil Pengujian IR Led Trasnmitter	39
4.2.3	Hasil Pengujian Perangkat Lunak	42
4.3	Perbaikan	43
4.3.1	Perbaikan flow node Switch	43
4.3.2	Perbaikan IR Led Transmitter	44
4.3.3	Rangkaian Elektronik Perbaikan	46
4.3.4	Desain 3D perbaikan	46
4.3.5	Hasil Perbaikan Prototype	47
4.4	Hasil Pengujian Perbaikan	48
4.4.1	Hasil Pengujian flow node Switch.....	48

4.4.2 Hasil Pengujian Perbaikan IR Led Transmitter.....	49
Bab 5 Penutup.....	52
5.1 Kesimpulan	52
5.2 Saran Perbaikan	52
Daftar Pustaka	53
Lampiran Program ESP32.....	56



DAFTAR TABEL

Tabel 2-1 Perbandingan Mikrokontroler Board	10
Tabel 4-4 Hasil pengujian perangkat lunak NodeRED	42



DAFTAR GAMBAR

Gambar 2-1 Tampilan NodeRED pada <i>Command Prompt Windows</i>	7
Gambar 2-2 Workspace NodeRed	8
Gambar 2-3 MQTT	9
Gambar 2-4 Logo Mosquitto	10
Gambar 2-5 ESP32 Board	10
Gambar 2-6 PIR sensor HC-SR501	11
Gambar 2-7 IR LED Transmitter	12
Gambar 3-1 Diagram alir penelitian	13
Gambar 3-2 Diagram sistem manual input	15
Gambar 3-3 Diagram sistem otomatisasi	16
Gambar 3-4 Diagram alir sistem 1	17
Gambar 3-5 Diagram alir sistem 2	18
Gambar 3-6 Diagram Publish MQTT	19
Gambar 3-7 Diagram Subscribe MQTT	19
Gambar 3-8 Diagram alir program ESP32	20
Gambar 3-9 Diagram alir program ESP32 Switch	21
Gambar 3-10 Diagram alir program ESP32 Set Suhu	22
Gambar 3-11 Diagram alir program ESP32 Fan AC	23
Gambar 3-12 Library IRremoteESP8266	23
Gambar 3-13 Isi contoh program-program dari library IRremoteESP8266	24
Gambar 3-14 Contoh protokol kontrol AC Daikin	24
Gambar 3-15 Flow Node Switch On-Off	25
Gambar 3-16 Flow Node Set Suhu	25
Gambar 3-17 Flow Node Set Fan AC	26
Gambar 3-18 Flow Node monitoring ruangan	26
Gambar 3-19 Desain rangkaian elektronik alat	26
Gambar 3-20 Rangkaian elektronik alat	27
Gambar 3-21 Desain 3D alat	28
Gambar 4-1 Serial monitor ESP32	35
Gambar 4-2 Comand prompt aktifitas NodeRED	36

Gambar 4-3 Contoh IP Adress NodeRED pada browser <i>smartphone</i>	36
Gambar 4-4 Pandangan atas posisi pengujian	37
Gambar 4-5 Posisi regulator full	37
Gambar 4-6 Posisi jumper pada H.....	37
Gambar 4-7 Grafik pengujian sensor PIR Sudut 0°	38
Gambar 4-8 Grafik pengujian sensor PIR Sudut 30°	38
Gambar 4-9 Grafik pengujian sensor PIR Sudut 60°	38
Gambar 4-10 Grafik pengujian sensor PIR Sudut 80°	39
Gambar 4-11 Pandangan atas posisi alat	39
Gambar 4-12 Pandangan depan posisi alat.....	40
Gambar 4-13 Hasil pengujian IR Led Sejar AC.....	40
Gambar 4-14 Pandangan atas posisi alat	40
Gambar 4-15 Pandangan depan posisi alat.....	41
Gambar 4-16 Hasil pengujian IR Led 2 AC pada AC 1.....	41
Gambar 4-17 Hasil pengujian IR Led 2 AC pada AC 2.....	41
Gambar 4-18 Dashboard NodeRED tampilan browser PC/Laptop.....	42
Gambar 4-19 Dashboard NodeRED tampilan browser <i>smartphone</i>	42
Gambar 4-20 Penambahan node trigger	43
Gambar 4-21 Penjelasan node trigger	43
Gambar 4-22 PSU.....	44
Gambar 4-23 Desain rangkaian IR Led Trasmmitter	45
Gambar 4-24 Rangkaian IR Led Transmitter perbaikan	45
Gambar 4-25 Rangkaian elektronik alat perbaikan	46
Gambar 4-26 Desain 3D perbaikan	46
Gambar 4-27 Prototipe	47
Gambar 4-28 Isi alat	47
Gambar 4-29 Posisi kedua AC diruang CNC.....	48
Gambar 4-30 Pandangan atas posisi alat	49
Gambar 4-31 Pandangan depan posisi alat.....	49
Gambar 4-32 Hasil perbaikan IR Led Transmitter 1 pada AC 1.....	49
Gambar 4-33 Hasil perbaikan IR Led Transmitter 1 pada AC 2.....	50
Gambar 4-34 Meninggikan posisi alat.....	50

Gambar 4-35 Hasil perbaikan IR Led Transmitter 2 pada AC 1..... 51
Gambar 4-36 Hasil perbaikan IR Led Transmitter 2 pada AC 2..... 51



DAFTAR NOTASI



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Air Conditioner (AC) saat ini sangat dibutuhkan tidak terkecuali pada ruangan baik kelas, laboratorium dan ruangan lainnya yang ada pada kampus Universitas Islam Indonesia. Pengkondisian udara sangat dibutuhkan dalam proses pembelajaran untuk memberikan kenyamanan baik bagi mahasiswa maupun dosen pengampu. Mengutip dari SNI Termal, suhu Sejuk-Nyaman berkisar antara 20.5°C-22.8°C (Pahrurrozi, 2020).

AC yang biasa digunakan di kampus Universitas Islam Indonesia khususnya di Fakultas Teknologi Industri, adalah tipe AC split. AC split adalah AC yang evaporator dan kondensornya berada ditempat yang terpisah. Dalam setiap ruangan kelas, biasanya terdapat 2 buah AC. Posisi dari AC tidak selalu pasti dalam setiap ruangan. Terkadang posisi AC depan dan belakang berhadapan, ada juga yang sejajar disamping saja atau didepan saja. Posisi pemasangan AC ditentukan dari mudahnya pemasangan (instalasi) dan aliran udara dari AC tersebut ke dalam ruangan.

Dalam satu ruangan, terkadang ditemukan pula yang menggunakan merk AC yang berbeda. Sehingga dalam satu ruangan terdapat dua remote pengendali AC. Remote AC kebanyakan menggunakan cahaya Inframerah untuk menyalurkan perintah-perintah pengendali AC. Remote AC biasanya diletakkan portable ditempat yang mudah dijangkau oleh pengguna AC. Remote AC yang peletakkannya sembarangan ini terkadang membuat remote AC hilang dari ruangan, atau dipinjam ke ruangan yang lain yang merk AC nya sama.

Sering kali ditemukan, ketika kegiatan perkuliahan dalam suatu ruang kelas telah usai, AC dalam kondisi menyala. Kejadian seperti ini biasa disebabkan karena kelalaian dari pengguna ruangan. Remote control AC yang menghilang dari ruangan juga salah satu sebab kejadian tersebut. AC yang menyala ketika ruangan tidak digunakan ini akan membuat pemborosan listrik dan juga pemborosan energi yang percuma.

Setiap fakultas di lingkungan kampus Universitas Islam Indonesia, sudah terkoneksi dengan jaringan wifi. Jaringan wifi ini mampu digunakan untuk membuat suatu alat dengan memanfaatkan Internet of Things (IoT) yaitu suatu konsep sistem yang berupaya untuk mengintegrasikan dan menghubungkan semua perangkat elektronik dengan jaringan internet (Pradana & Nurfiana, 2019). Alat yang akan dibuat diharapkan mampu untuk mengontrol penggunaan AC secara otomatis sehingga pemborosan akibat AC yang tidak digunakan ketika tidak ada kegiatan dalam suatu ruangan mampu dikurangi.

1.2 Rumusan Masalah

Perumusan masalah yang akan dibahas pada proposal ini adalah :

1. Bagaimana agar remote tidak hilang?
2. Bagaimana satu remote mampu mengontrol 2 tipe AC yang berbeda?
3. Bagaimana agar AC dapat dikontrol dari mana saja dengan koneksi internet?
4. Bagaimana agar menghentikan pemborosan AC yang menyala ketika ruangan sudah tidak terpakai?
5. Bagaimana agar AC mampu memberikan suhu nyaman yang terbaik?

1.3 Batasan Masalah

Pembatasan masalah dalam penelitian ini agar ruang lingkup pembahasan menjadi jelas dan tidak meluas ke hal-hal yang tidak diinginkan. Pembatasan masalah dalam penelitian ini meliputi hal-hal sebagai berikut :

1. Hasil penelitian berupa prototipe.
2. Merk AC yang dilakukan untuk penelitian disini hanya merk Daikin.
3. Ruangan yang digunakan untuk penelitian adalah ruang training Laboratorium Mekatronika dan juga ruang CNC di lantai bawah gedung KH Wahid Hasyim.
4. Jarak antara 2 buah AC 6 meter, dengan tinggi diukur dari tanah 2.8 meter.
5. Material pembuatan prototipe menggunakan PLA dengan proses 3D print.

6. Otomatisasi dengan IoT yang dapat dilakukan yaitu mampu menghidupkan dan mematikan AC, mengontrol suhu AC secara otomatis sesuai kondisi udara yang terdeteksi pada ruangan.
7. Desain prototype menggunakan software Solidworks 2014.

1.4 Tujuan Penelitian atau Perancangan

Penelitian ini bertujuan untuk merancang sistem dan alat untuk menggantikan remote AC konvensional, dimana alat ini mampu mengontrol lebih dari satu AC pada suatu ruangan dengan memanfaatkan koneksi internet dan IOT, ditambah juga dengan tampilan antar-muka bagi pengguna yang bisa digunakan untuk memonitoring ruangan dan kontrol manual.

1.5 Manfaat Penelitian atau Perancangan

Manfaat penelitian ini dibagi menjadi beberapa kepentingan, yaitu :

1.5.1 Manfaat Penelitian bagi Mahasiswa

- a. Meningkatkan kemampuan mahasiswa dalam mengaplikasikan teori yang telah didapatkan di bangku perkuliahan.
- b. Dapat memberikan pemahaman yang lebih dalam bagi mahasiswa mengenai pembuatan sistem berbasis IoT.
- c. Mahasiswa dapat mengetahui permasalahan yang terjadi pada pembuatan dan perancangan alat dan sistem yang dirancang.

1.5.2 Manfaat Penelitian bagi Universitas

- a. Alat dapat diaplikasikan pada ruangan-ruangan di kampus.
- b. Pengontrolan AC pada suatu ruangan kelas bisa dilakukan dari mana saja dengan koneksi internet.
- c. Mengurangi pemborosan yang terjadi karena penggunaan AC.

1.6 Sistematika Penulisan

Sistematika penulisan Proposal Tugas Akhir ini dijelaskan sebagai berikut.

1.6.1 BAB I – Pendahuluan

Bab ini menjelaskan latar belakang dari masalah yang akan diangkat dan dibahas pada proposal ini.

1.6.2 BAB II – Tinjauan Pustaka

Bab ini berisi tentang kajian pustaka dan dasar teori yang akan digunakan dalam penelitian dan perancangan.

1.6.3 BAB III – Metode Penelitian

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian dan metode penelitian yang digunakan.

1.6.4 BAB IV – Hasil dan Pembahasan

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian dan metode penelitian yang digunakan.

1.6.5 BAB V – Penutup

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian dan metode penelitian yang digunakan.

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Berikut beberapa kajian pustaka dari penelitian-penelitian yang sudah dilakukan sebelumnya sebagai bahan acuan bagi penelitian kali ini.

Pada jurnal milik Natsir dkk (2019), peneliti menggunakan mikrokontroler Arduino Mega 2560 dan ditambah dengan Ethernet Shield W5500 untuk mengkoneksikan Arduino ke internet dengan kabel RJ-45. Sistem bekerja dengan input dari sensor Ultrasonik yang menghitung jumlah orang yang masuk ruangan dan juga suhu ruangan yang didapat dari sensor suhu DS18B20, kemudian data akan digunakan untuk set-suhu secara otomatis oleh Arduino. Arduino kemudian mengirimkan sinyal IR untuk mengontrol AC. Data ini juga kemudian akan dikirim oleh Ethernet Shield ke webserver dengan user interface melalui browser dan LCD 20x4 yang menampilkan berapa orang yang masuk ruangan dan suhu ruangan. Sistem juga dilengkapi dengan buzzer dan led sebagai counter pengunjung ruangan.

Jurnal yang lain yaitu milik Alvan Prastoyo Utomo dkk (2019), hampir sama dengan jurnal pertama diatas. Perbedaannya ada pada sensor input yang digunakan. Jurnal milik Natsir dkk (2019) menggunakan sensor Ultrasonik sebagai counter jumlah pengunjung ruangan, sedangkan jurnal ini hanya berfokus pada otomatisasi setting suhu AC karena penggunaan sistem hanya pada ruangan server. Sensor suhu juga berbeda yaitu menggunakan DHT22. Arduino Mega digunakan hanya sebagai aktuator untuk mengirim sinyal IR ke AC, sedangkan untuk koneksi internet menggunakan Raspberry Pi. Pada jurnal ini digunakan juga database untuk menyimpan data suhu dan kelembaban ruangan menggunakan MongoDB. Jadi output dari sistem akan bisa dilihat pada suatu tampilan webserver dan juga dengan aplikasi chatting Telegram.

Jurnal selanjutnya yaitu milik Diori dkk (2019), peneliti bertujuan membuat sistem yang mampu mematikan dan menyalakan AC otomatis dan mampu mendeteksi kerusakan yang terjadi pada AC sekaligus memberi peringatan

maintenance bagi pengguna. Sensor yang digunakan berbeda dengan milik Natsir dkk (2019). Inputnya menggunakan sensor PIR untuk mendeteksi gerakan pada suatu ruangan. Mikrokontroler yang digunakan menggunakan 2 buah yaitu Arduino Uno dan NodeMCU. Karena sistem mampu mendeteksi kerusakan juga, ditambahkan sensor water level pada tandon air AC dan sensor hall effect sebagai indikator putaran kipas pada evaporator dan kondenser AC. Peneliti menggunakan platform *thingspeak.com* sebagai penyimpanan data dari sensor.

Jurnal selanjutnya yaitu milik Pahrurrozi (2020) juga sama dengan ketiga jurnal diatas. Peneliti membuat suatu sistem yang mampu mengendalikan AC secara otomatis menggunakan mikrokontroler Wemos D1 R2. Sensor input menggunakan PIR sensor dan juga sensor suhu untuk memonitoring ruangan menggunakan DHT22. Output ke AC sama dengan ketiga modul diatas yaitu infrared (IR). Jurnal ini juga menggunakan MQTT sebagai protokol komunikasi untuk IoT dan menggunakan webserver MAC PSTI.

Jurnal selanjutnya yaitu jurnal milik Periyaldi dkk (2018), lebih mirip dengan jurnal milik Diori dkk (2019) karena fungsi dari sistem yang dibuat hanya memonitoring suhu suatu ruang server. Pada jurnal ini, penulis menggunakan mikrokontroler Raspberry Pi yang disambungkan dengan sensor input suhu DS18B20. Pada jurnal ini dijelaskan pula tentang broker MQTT yang digunakan yaitu Mosquitto dan juga software web service yaitu NodeRED. NodeRED digunakan sebagai sarana antarmuka dalam memonitoring ruangan dengan tampilan dashboard realtimenya.

Dari seluruh jurnal diatas, semuanya memanfaatkan IoT. Setiap penelitian pasti menggunakan mikrokontroler, sebagai contoh Arduino. Penggunaan Arduino ditambah dengan *Ethernet shield* menurut saya kurang efisien dan memakan tempat, karena menjadikan Arduino hanya sebagai aktuator dan *Ethernet shield* hanya untuk menyambungkan Arduino ke internet. Sudah tersedia berbagai mikrokontroler yang sudah memiliki Wifi modul didalamnya misal seperti NodeMCU dan ESP32.

Jurnal-jurnal diatas menggunakan sensor suhu seperti DHT22 dan DS18B20. Menurut saya kemampuan kedua sensor tersebut berlebihan ketika hanya untuk mengukur suhu suatu ruangan. DHT11 sudah cukup untuk mengukur

suhu dan kelembapan ruangan dan harganya lebih murah dibandingkan kedua sensor yang telah dipakai pada jurnal diatas.

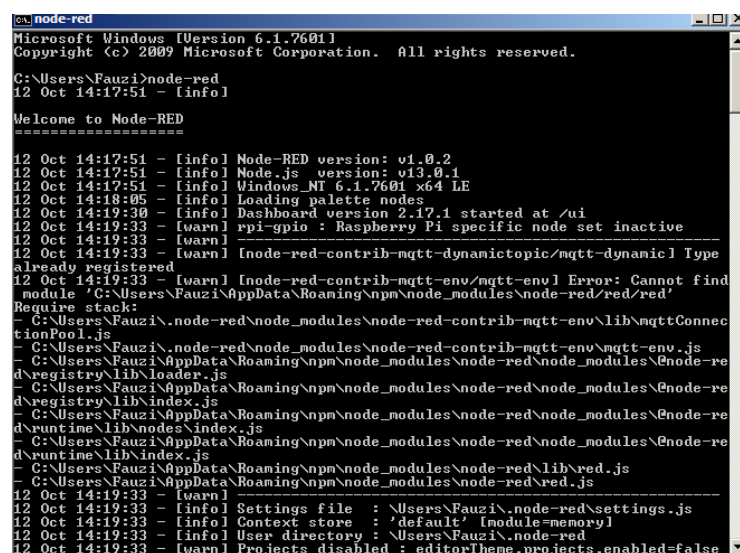
Pada jurnal milik Natsir dkk (2019), peneliti masih menggunakan LCD sebagai monitor manualnya. Menurut saya ini kurang efisien karena hanya bisa digunakan pada satu tempat, dan kurang mobilitasnya. Penggunaan webserver yang mampu menampilkan antar-muka yang bisa dibuka menggunakan smartphone ataupun gawai yang sehari-hari dipakai lebih nyaman dan meningkatkan mobilitas.

Penelitian ini ditujukan untuk memonitoring kondisi dari AC pada suatu ruangan. Dengan memanfaatkan IoT, sangat memudahkan dalam memonitoring suatu perangkat atau lingkungan bahkan dari tempat yang jauh selama seluruh perangkat terkoneksi internet. Sejatinya IoT, memberikan kemudahan koneksi dan komunikasi antara objek fisik atau suatu sistem dan manusia (Ahmadi, 2019).

2.2 Dasar Teori

2.2.1 NodeRed

NodeRed adalah suatu alat pemrograman untuk mengoneksikan beberapa perangkat, API dan online service menjadi bagian dari Internet of Things. NodeRed dikembangkan oleh perusahaan IBM. Pada 2016, IBM menjadikan NodeRed sebagai bagian dari JS Foundation open source projek (Rafly, 2020).



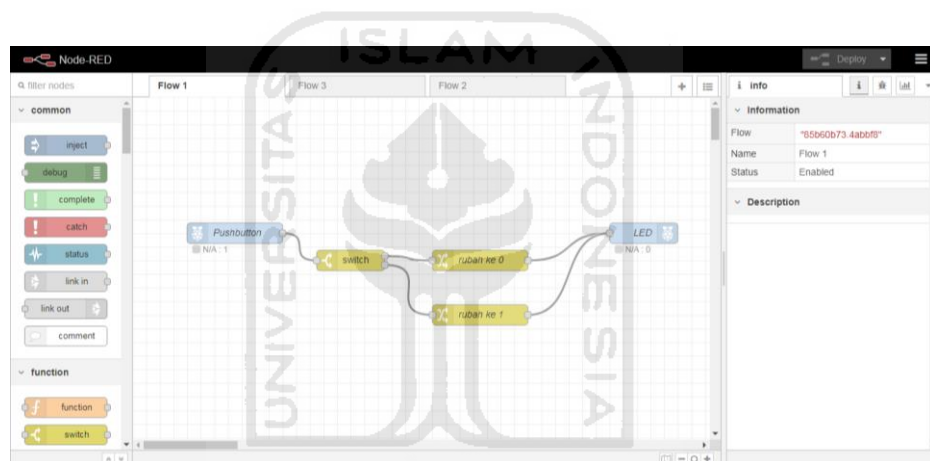
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Fauzi>node-red
12 Oct 14:17:51 - [Info]

Welcome to Node-RED
=====
12 Oct 14:17:51 - [Info] Node-RED version: v1.0.2
12 Oct 14:17:51 - [Info] Node.js version: v13.0.1
12 Oct 14:17:51 - [Info] Windows_NT 6.1.7601 x64 LE
12 Oct 14:18:05 - [Info] Loading palette nodes
12 Oct 14:19:30 - [Info] Dashboard version 2.17.1 started at /ui
12 Oct 14:19:33 - [Warn] rpi-gpio : Raspberry Pi specific node set inactive
12 Oct 14:19:33 - [Warn] [node-red-contrib-nqtt-dynanictopic/nqtt-dynamic] Type
already registered
12 Oct 14:19:33 - [Warn] [node-red-contrib-nqtt-env/nqtt-env] Error: Cannot find
module 'C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\red\red'
Require stack:
- C:\Users\Fauzi\node-red\node_modules\node-red-contrib-nqtt-env\lib\nqttConnec
tionPool.js
- C:\Users\Fauzi\node-red\node_modules\node-red-contrib-nqtt-env\nqtt-env.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\node_modules\node-re
d\nregistry\lib\loader.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\node_modules\node-re
d\nregistry\lib\index.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\node_modules\node-re
d\nruntime\lib\nodes\index.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\node_modules\node-re
d\nruntime\lib\index.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\lib\red.js
- C:\Users\Fauzi\AppData\Roaming\npm\node_modules\node-red\red.js
12 Oct 14:19:33 - [Warn]
12 Oct 14:19:33 - [Info] Settings file : \Users\Fauzi\node-red\settings.js
12 Oct 14:19:33 - [Info] Context store : default | In-memory
12 Oct 14:19:33 - [Info] User directory : \Users\Fauzi\node-red
12 Oct 14:19:33 - [Warn] Projects disabled : editorTheme.projects.enabled=false
```

Gambar 2-1 Tampilan NodeRED pada *Command Prompt Windows*

Tampilan Nodered dibuka pada command prompt windows maupun terminal. Terlihat pada gambar 2-1 tampilan NodeRED pada command prompt windows. Untuk mempermudah dalam pemrograman, NodeRed menggunakan basis browser-based flow editing, dimana kita menggunakan web browser sebagai sarana pemrograman. NodeRed memiliki palette yang berisi fungsi-fungsi (disebut nodes) yang bisa digabungkan menjadi suatu flow program atau perintah. Pemrograman Function nodes dilakukan dengan menggunakan pemrograman JavaScript. NodeRed menjadikan pemrograman menjadi lebih mudah dan visual terutama bagi kalangan yang awam tentang berbagai macam jenis pemrograman. NodeRED berfokus ke program berbentuk *flow*, dimana *flow* tersebut dibuat dari bermacam-macam *node-node* yang saling dihubungkan (Rafly, 2020).

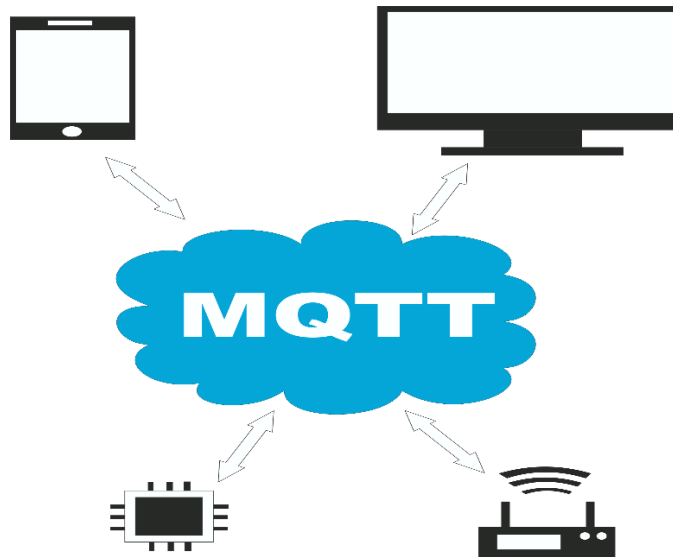


Gambar 2-2 Workspace NodeRed

NodeRed tersedia untuk berbagai Operating System seperti Windows, iOS dan juga Linux. NodeRed juga sudah tersedia built-in pada Raspberry Pi.

2.2.2 MQTT

MQTT atau Message Queue Telemetry Transport adalah protokol yang sangat sering digunakan dalam pembuatan proyek IoT. Menggunakan mekanisme *publish* atau *subscribe*, suatu alat dapat *subscribe* suatu topik tertentu. Sehingga apabila server biasa disebut broker pada MQTT mempublish suatu message/pesan pada topik yang telah disubscribe oleh suatu alat, maka alat akan menerima pesan yang dikirimkan dari server (Andreas et al., 2019).



Gambar 2-3 MQTT

Ada 3 bagian penting pada MQTT, yaitu *publisher*, *subscriber* dan broker. MQTT dijalankan dengan menggunakan suatu broker penyedia layanan MQTT. alat akan terhubung pada broker yang memiliki topik tertentu.berikut penjelasan masing-masing bagian.

1. Publish digunakan oleh suatu alat untuk mengirimkan data ke *subscriber*. Pada tahap publish biasanya alat berhubungan dengan sensor tertentu. *Publisher* adalah alat yang mengirimkan data ke *subscriber*.
2. Subscribe kebalikan dari publish dimana subscribe akan menerima data dari *publisher*.alat yang menerima data ini disebut *subscriber*.
3. Broker berfungsi sebagai jembatan antara *publisher* dan *subscriber* data.
4. Topik disini adalah suatu judul dari data tersebut agar lebih terkategori. *Publisher* dan *subscriber* harus berada pada topik yang sama agar data tersampaikan.(Adyawadhana, n.d.,2020)

MQTT pada penelitian ini akan digunakan sebagai protokol komunikasi antara ESP32, sensor dan NodeRED. ESP32 dan NodeRED keduanya menjadi *publisher* sekaligus juga *subscriber*. Karena masing-masing alat mengirim dan menerima data dari beberapa topik yang sama. MQTT sangat cocok digunakan dalam suatu porjek IoT karena ukuran data yang kecil, penggunaan daya yang rendah dan mudah diimplementasikan. Broker yang dipilih sebagai server MQTT

menggunakan Mosquitto. Mosquitto adalah broker MQTT yang sifatnya open-source sehingga gratis dan akan terus dikembangkan (Periyaldi et al., 2018).



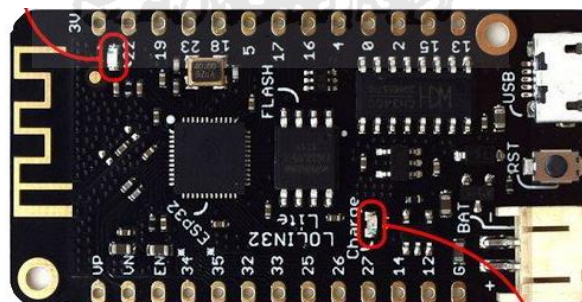
Gambar 2-4 Logo Mosquitto

(sumber : <https://mosquitto.org/images/mosquitto-text-side-28.png>)

2.2.3 ESP32

ESP32 adalah salah satu mikrokontroler. ESP32 memiliki beberapa kelebihan dibanding mikrokontroler yang lain. Yang paling utama adalah *built-in* WiFi module dan Bluetooth yang memudahkan ESP32 dalam koneksi internet (Soelistijorini, 2019). Koneksi internet sangat dibutuhkan dalam suatu proyek IoT. ESP32 juga dilengkapi dengan GPIO Pin yang bisa digunakan untuk berbagai kebutuhan. Tersedia pula pin I2C dan SPI yang memudahkan untuk komunikasi.

Pemrograman ESP32 pada penelitian ini dilakukan menggunakan software Arduino IDE. ESP32 nantinya digunakan untuk otak dari alat penelitian. ESP32 juga berfungsi sebagai MQTT client untuk mem-*publish* dan juga men-*subscribe* topik-topik yang diperlukan dalam sistem..



Gambar 2-5 ESP32 Board

Tabel 2-1 Perbandingan Mikrokontroler Board

Spesifikasi	Arduino	ESP8266	ESP32
Clock Speed	16 MHz	80 MHz	160 MHz

Operating Voltage	5 V	3.3 V	3.3 V
GPIO	20	17	36
SRAM	2 kB	160 kB	512 kB
SPI	Ada	Ada	Ada
I2C	Ada	Ada	Ada
WiFi	-	HT20	HT40
Bluetooth	-	-	Bluetooth 4.2

2.2.4 PIR Sensor

PIR (Passive Infra Red) sensor adalah sensor yang sensitif terhadap sinar Infra merah (Mukhopadhyay et al., 2018). PIR sensor biasanya digunakan untuk pendeteksi gerakan. PIR sensor dilengkapi dengan lensa Fresnel, yang bertujuan untuk memfokuskan cahaya Infra merah dari berbagai arah ke sensor.



Gambar 2-6 PIR sensor HC-SR501

PIR sensor pada penelitian ini digunakan sebagai sensor pendeteksi gerakan pada suatu ruangan yang menggunakan AC. PIR akan memberi sinyal untuk mengaktifkan dan menonaktifkan AC secara otomatis.

2.2.5 IR LED (Infra Red LED)

IR LED adalah LED yang memancarkan sinar infra merah. Terbuat dari bahan Galium Arsenida (GaAs) yang dapat memancarkan cahaya infra merah dan juga radiasi panas ketika diberi aliran listrik (Aksin M., 2013). IR LED terdiri dari 2 macam, sebagai transmitter dan receiver.



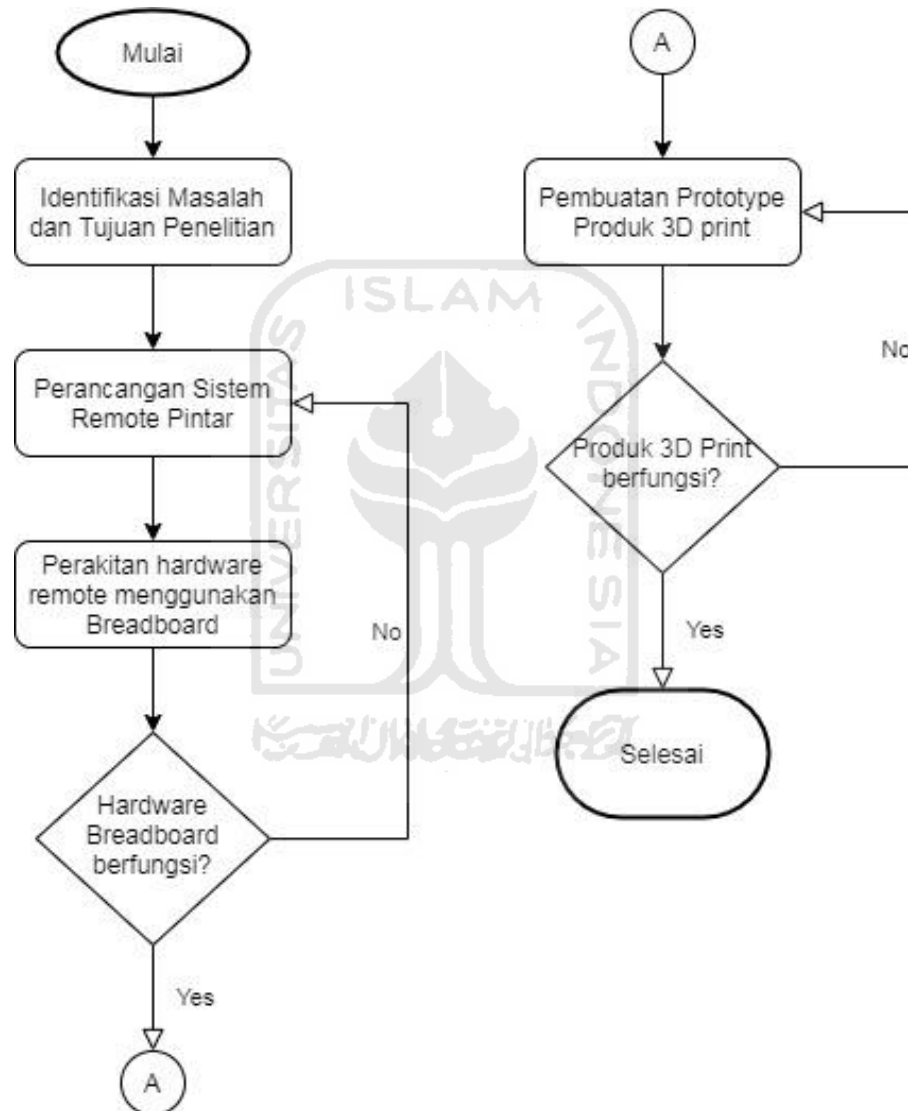
Gambar 2-7 IR LED Transmitter

IR LED pada penelitian ini yang akan digunakan adalah IR LED transmitter. IR LED yang digunakan ukuran 5mm. Digunakan untuk mengirimkan sinyal perintah untuk mengontrol AC.

BAB 3 METODE PENELITIAN

3.1 Alur Penelitian

Diagram alir penelitian ini dari beberapa tahap penelitian yang dilakukan dapat dilihat dalam bagan Gambar 3.1 berikut.



Gambar 3-1 Diagram alir penelitian

3.2 Peralatan dan Bahan

Dalam perancangan alat kontrol AC otomatis berbasis IoT ini, ada beberapa peralatan dan bahan yang dibutuhkan, yaitu:

3.2.1 Perangkat Keras

- Laptop/PC untuk pembuatan kode aplikasi dan untuk penggunaan NodeRED
- AC Daikin yang terletak pada ruang training Laboratorium Mekatronika dan ruang CNC Prodi Teknik Mesin UII
- Soldir
- 3D Printer Ender Pro 3.5

3.2.2 Perangkat Lunak

- Arduino IDE untuk membuat program ESP32
- Chrome Browser untuk mengakses NodeRED
- Command Prompt untuk mengaktifkan NodeRED
- Solidworks untuk membuat desain 3D alat
- Ultimaker Cura untuk proses slicing dalam pembuatan 3D print

3.2.3 Bahan

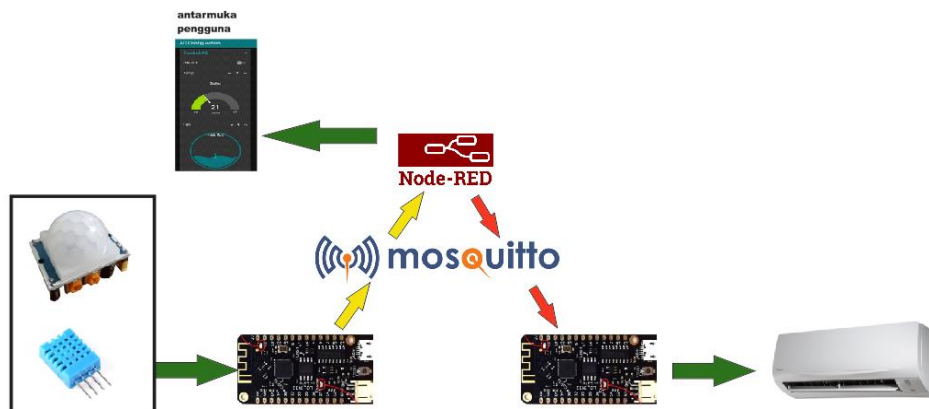
- 1 buah mikrokontroller ESP32
- 1 buah sensor suhu DHT11 sebagai pendeteksi suhu ruangan
- 1 buah sensor PIR HC-SR501 sebagai pendeteksi gerakan
- 2 buah IR Led sebagai transmitter sinyal Infrared
- Breadboard
- PCB bolong
- Kabel jumper
- Timah
- 1 buah PSU 5v
- 4 buah batre A2

- Box batre A2
- Transistor NPN 2N2222a

3.3 Perancangan Sistem Alat

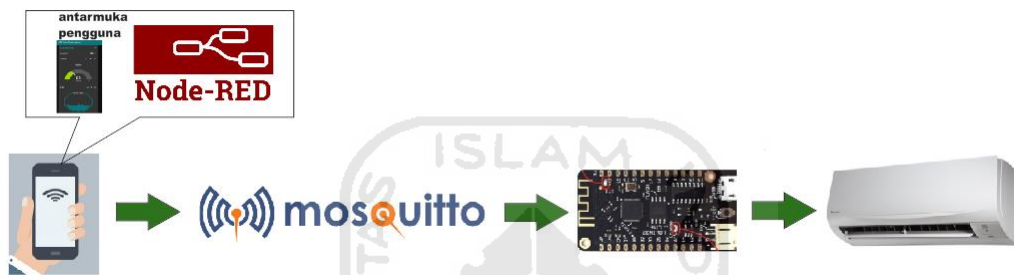
Alat dibagi menjadi 2 yaitu untuk sebagai server dan sebagai transmitter ke AC. Server disini digunakan NodeRED yang terinstall pada Laptop. NodeRED digunakan karena kemudahan pemrogramannya, dan juga NodeRED mampu menggambarkan hubungan-hubungan dari MQTT seperti *publisher* dan *subscriber* secara jelas dan mudah dengan cara menyambungkan antar node-node yang tersedia pada NodeRED. Transmitter ke AC menggunakan mikrokontroler ESP32 yang mampu terkoneksi dengan internet melalui wifi. ESP32 juga sudah memiliki library untuk MQTT pada Arduino, sehingga penggunaan tinggal memodifikasi program pada library tersebut.

Alat transmitter diharapkan mampu melakukan otomatisasi untuk menghidupkan dan mematikan AC, kemudian mampu mensetting suhu AC secara otomatis. Sensor PIR digunakan untuk mendeteksi gerakan pada ruangan yang dikontrol. Sensor suhu disini menggunakan DHT11 untuk memonitoring suhu ruangan secara langsung pada ruangan. Kedua sensor ini dihubungkan pada unit transmitter yaitu dengan mikrokontroler ESP32 seperti pada gambar 3-2. Data dari kedua sensor kemudian di-publish oleh ESP32 ke NodeRED sebagai server menggunakan MQTT yang akan ditampilkan pada tampilan antarmuka bagi pengguna melalui browser. NodeRED juga mengirimkan data kembali ke ESP32 untuk diproses menggunakan sinyal inframerah ke AC.

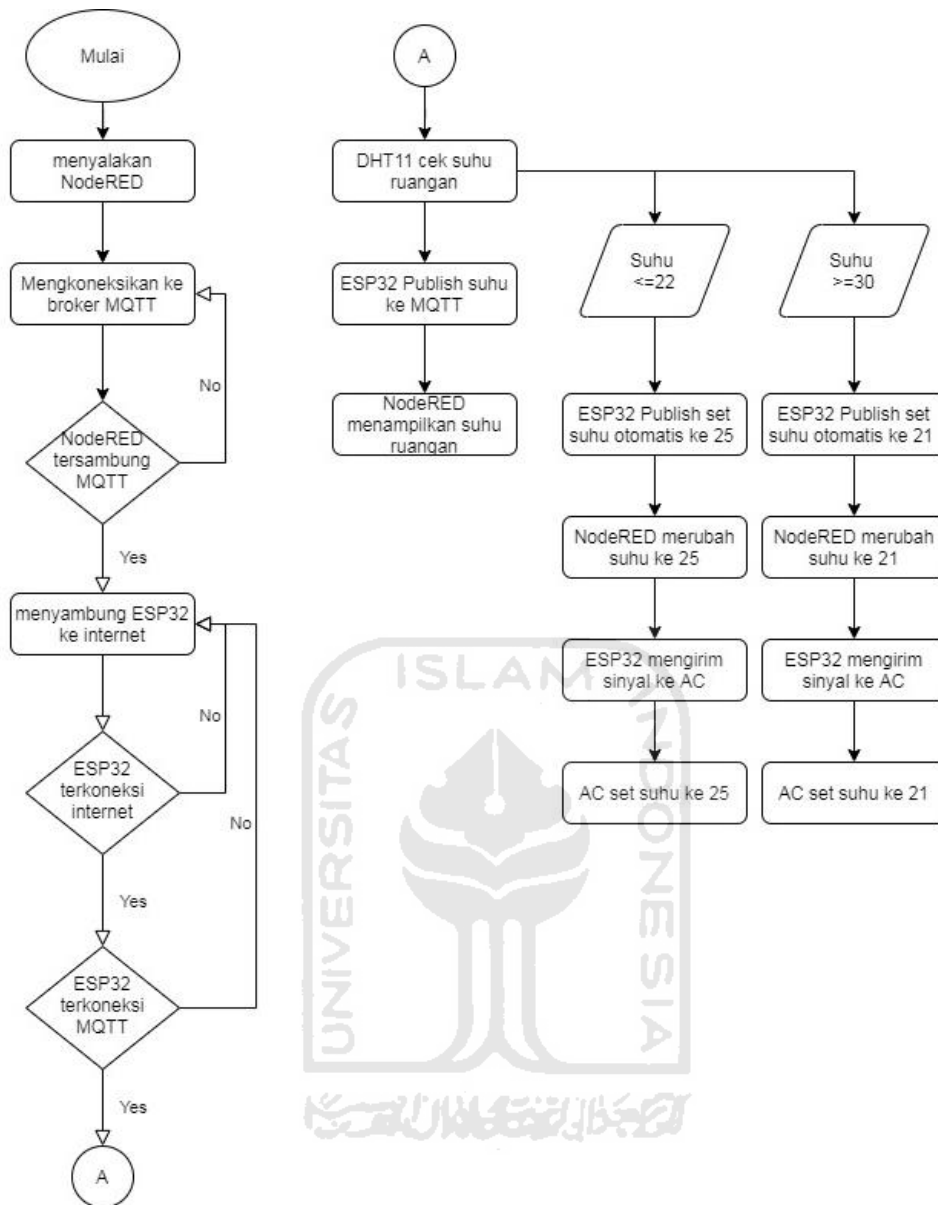


Gambar 3-2 Diagram sistem otomatisasi

Alat juga difungsikan sebagai remote kontrol AC manual, yang dapat diakses oleh setiap pengguna melalui tampilan antarmuka NodeRED yang bisa dibuka pada browser baik smartphone maupun laptop yang terkoneksi dengan internet yang sama dengan server NodeREDnya menggunakan ip address dari NodeRED. Dari kontrol manual pada tampilan antarmuka NodeRED tersebut kemudian diproses oleh NodeRED untuk mengirim data-data kontrol melalui MQTT ke ESP32 yang akan meneruskan menggunakan protokol inframerah ke AC. Penjelasannya singkatnya seperti gambar 3-3. Penjelasan lengkap bisa dilihat pada diagram alir 3-4 dan diagram alir 3-5.

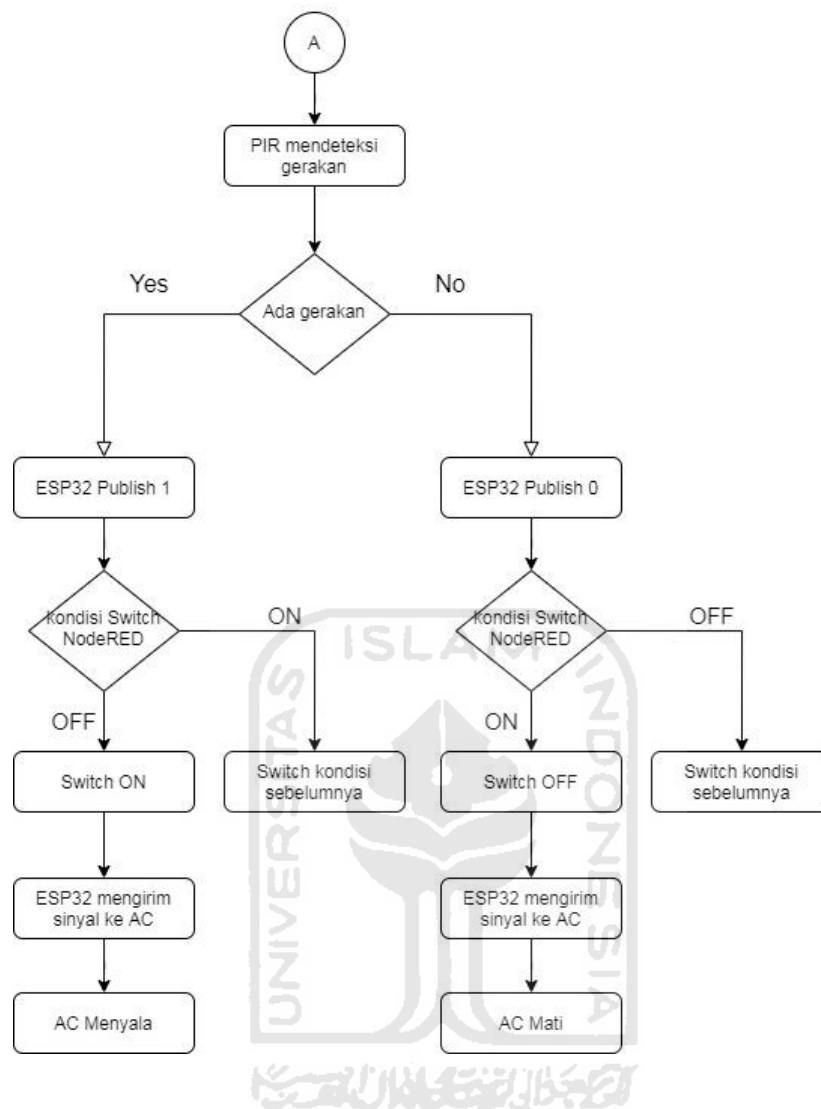


Gambar 3-3 Diagram sistem input manual



Gambar 3-4 Diagram alir sistem 1

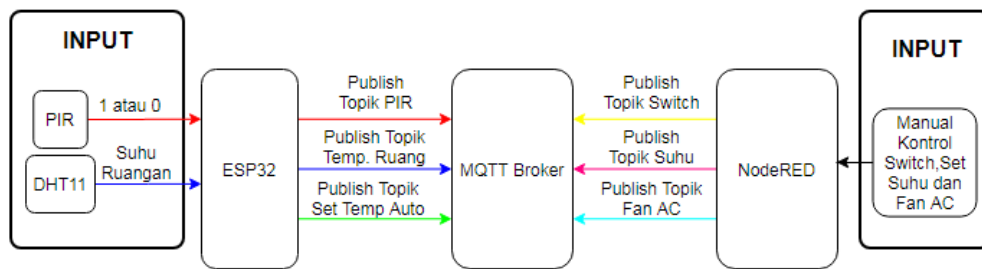
Baik ESP32 dan NodeRED dimulai dengan menyambungkan keduanya ke internet. DHT11 akan mengirimkan data suhu ruangan dengan publish ke MQTT broker dengan topik yang sama dengan NodeRED. Data suhu ruangan ditampilkan pada dashboard NodeRED. Suhu ruangan juga berfungsi untuk mengatur otomatis suhu settingan AC yang diperlukan. Ketika suhu ruangan lebih dari 30 derajat, maka AC akan diubah otomatis ke suhu 21. Kebalikannya, ketika suhu AC dibawah 22 derajat maka AC akan diset ke suhu 25 derajat.



Gambar 3-5 Diagram alir sistem 2

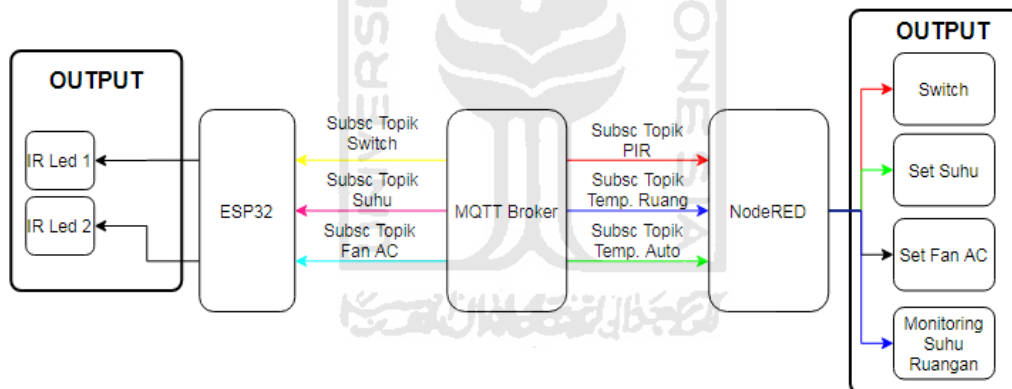
Pendeteksi gerakan menggunakan sensor PIR HC-SR501. Yang akan memberikan sinyal ketika ada gerakan AC menyala. Dan sebaliknya, ketika tidak terdeteksi gerakan pada ruangan, AC akan dimatikan otomatis.

3.4 Perancangan sistem Publish dan Subscribe MQTT



Gambar 3-6 Diagram Publish MQTT

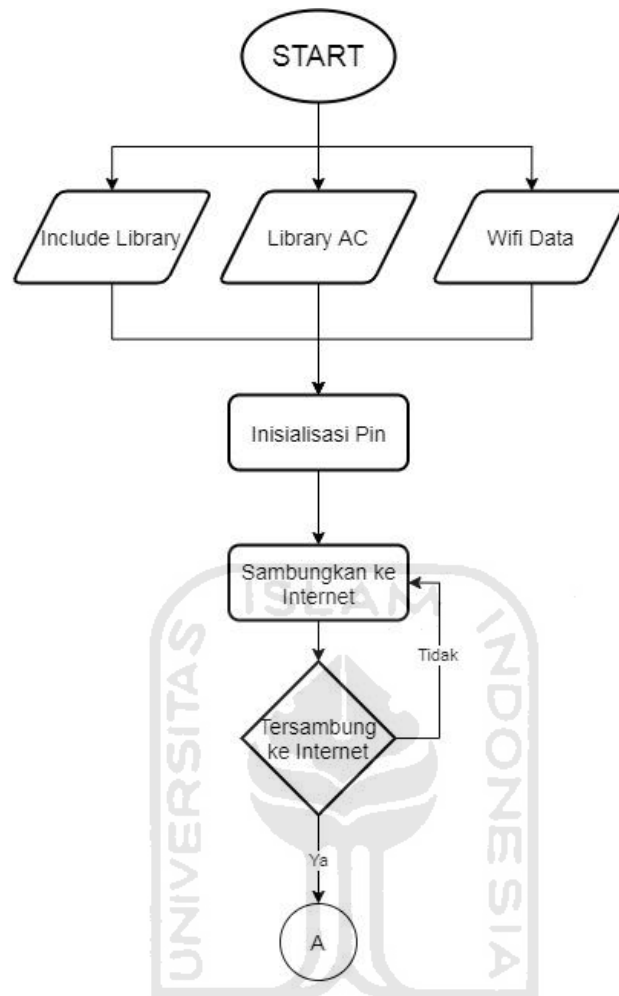
Gambar diatas menjelaskan tentang topik-topik yang di publish ke MQTT broker yaitu mosquitto. ESP32 mempublish topik-topik dari input sensor PIR dan DHT11. Sedangkan NodeRED mempublish topik-topik dari inputan manual kontrol seperti Switch on-off AC, Set suhu AC dan Fan AC. Ketiga kontrol manual ini difungsikan ketika pengguna ingin menggunakan remote secara manual.



Gambar 3-7 Diagram Subscribe MQTT

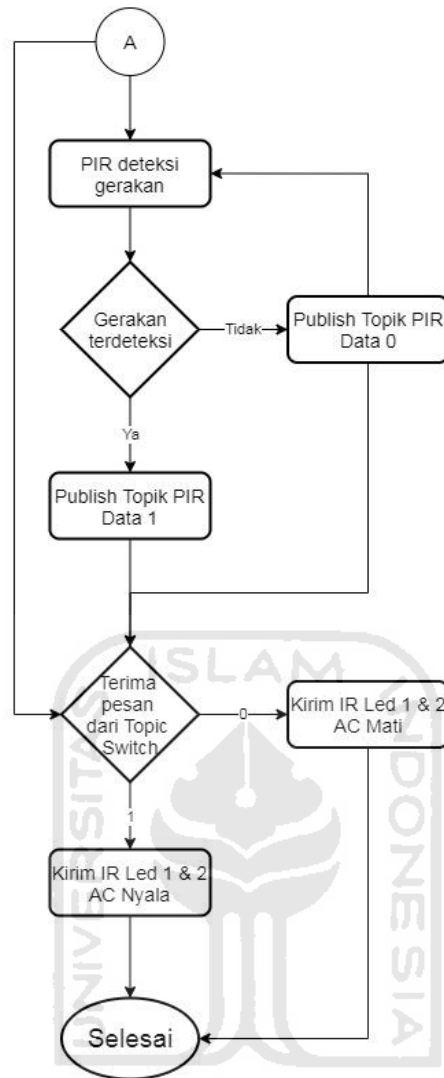
Gambar 3-4 memberikan gambaran tentang topik-topik yang disubscribe oleh ESP32 dan NodeRED. Perhatikan warna panah-panah pada setiap topik berhubungan dengan warna anak panah topik yang di publish pada gambar 2. Topik-topik yang disubscribe oleh ESP32 kemudian diteruskan ke output berupa perintah infra merah untuk mengendalikan AC. Output dari topik-topik yang disubscribe oleh NodeRED akan ditampilkan pada tampilan dashboard antar muka NodeRED, yang berisi Switch on-off, Set suhu AC dan Fan AC yang ketiganya bisa juga dikendalikan secara manual seperti gambar 2, dan yang terakhir yaitu monitoring suhu asli dari ruangan.

3.5 Perancangan Program ESP32



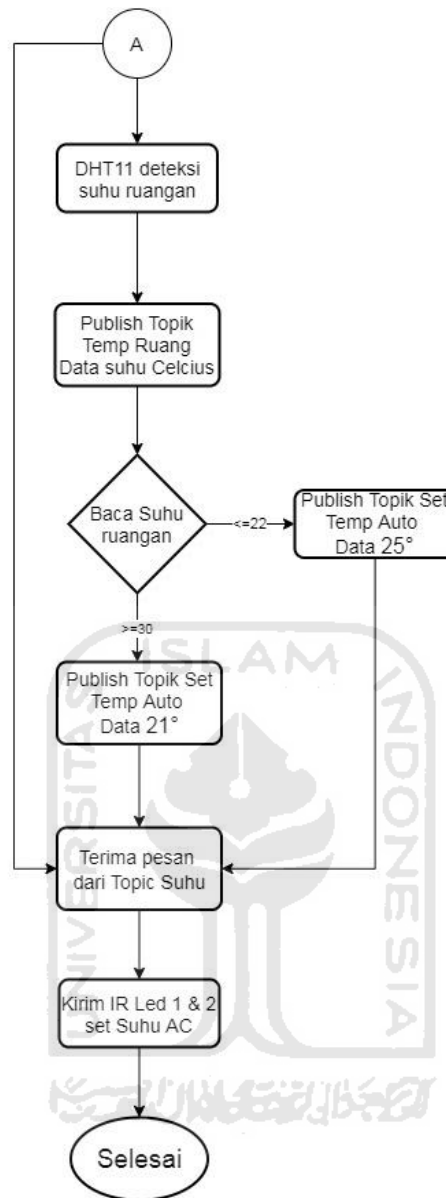
Gambar 3-8 Diagram alir program ESP32

Gambar 3-5 menjelaskan tentang awal flowchart program yang dituliskan pada software Arduino IDE untuk mikrokontroler ESP32. Program dimulai dari memasukkan data-data library yang akan digunakan dan juga data koneksi wifi seperti username dan password dan juga port mqtt yang dipakai. Data juga berupa AC yang digunakan dan protokol-protokol data yang akan dikirimkan ke AC seperti protokol on-off, setting suhu dan Fan AC yang sudah tersedia pada library Arduino. Program berlanjut ke inisialisasi pin-pin pada ESP32 yang akan digunakan untuk sensor PIR, DHT11, dan juga IR Led 1 dan 2. Program selanjutnya akan menyambungkan ESP32 ke internet melalui wifi.



Gambar 3-9 Diagram alir program ESP32 Switch

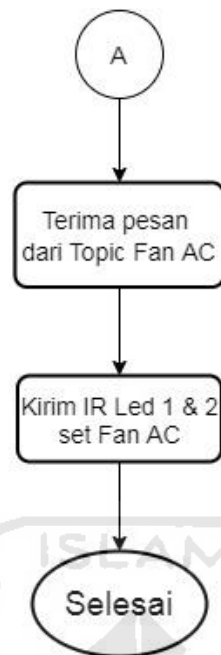
Gambar 3-6 adalah kelanjutan dari gambar 4. Ketika ESP32 telah terkoneksi internet maka otomatis protokol MQTT sudah tersambung, sehingga NodeRED dan ESP32 sudah tersambung. Switch on-off pada dashboard antar muka NodeRED bisa digunakan manual maupun otomatis karena sensor PIR seperti flowchart diatas. Ketika PIR mendeteksi gerakan maka ESP32 akan memublish ke broker MQTT dengan data string “1” pada topik PIR yang disubscribe oleh NodeRED. NodeRED akan memublish topik Switch dengan data 1 sesuai pesan dari ESP32. ESP32 pun akan menerima pesan dari topik Switch dari NodeRED sehingga IR Led mengirimkan sinyal perintah untuk menghidupkan AC.



Gambar 3-10 Diagram alir program ESP32 Set Suhu

Program set suhu ini bisa dilakukan secara otomatis maupun manual. DHT11 membaca suhu pada ruangan yang akan dipublish oleh ESP32 dengan topik Temp Ruang. NodeRED mensubscribe topik tersebut dan akan memunculkannya pada dashboard antar muka NodeRED. Ketika suhu ruangan lebih dari atau sama dengan 30°C maka ESP32 akan mempublish data yang akan mensetting suhu AC ke 21°C pada topik Set Temp Auto. NodeRED menerima pesan dari topik Set Temp Auto kemudian mempublish topik Suhu dengan data yang sama. ESP32 kemudian menerima pesan dari topik Suhu tersebut dan mengirimkannya ke AC. Begitu juga ketika suhu ruangan kurang dari atau sama

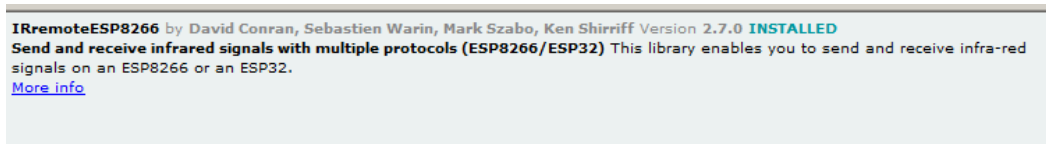
dengan 22°C maka dengan proses sama seperti yang telah dijelaskan AC akan diset secara otomatis ke suhu 25°C.



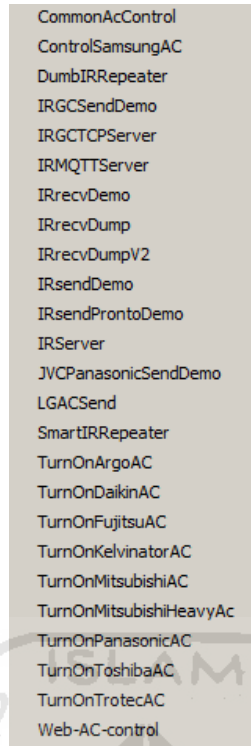
Gambar 3-11 Diagram alir program ESP32 Fan AC

Program yang terakhir ini tentang setting kecepatan AC, hanya dilakukan secara manual pada dashboard antar muka NodeRED. Program ini hanya berfungsi untuk mensetting kecepatan Fan AC.

Dalam pembuatan program pada ESP32, untuk protokol dari AC yang akan digunakan dipilih dari library yang sudah disediakan oleh Arduino. Bisa didownload pada library yaitu IRremoteESP8266. Pada library ini sudah tersedia berbagai merk dan tipe AC yang sering digunakan seperti yang ditunjukkan pada gambar 3-9 dibawah.



Gambar 3-12 Library IRremoteESP8266



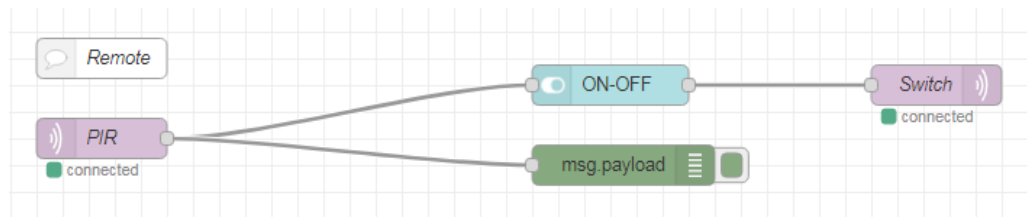
Gambar 3-13 Isi contoh program-program dari library IRremoteESP8266

```
39
40
41 void loop() {
42   Serial.println("Sending...");
43
44   // Set up what we want to send. See ir_Daikin.cpp for all
45   ac.on();
46   ac.setFan(1);
47   ac.setMode(kDaikinCool);
48   ac.setTemp(25);
49   ac.setSwingVertical(false);
50   ac.setSwingHorizontal(false);
51
```

Gambar 3-14 Contoh protokol kontrol AC Daikin

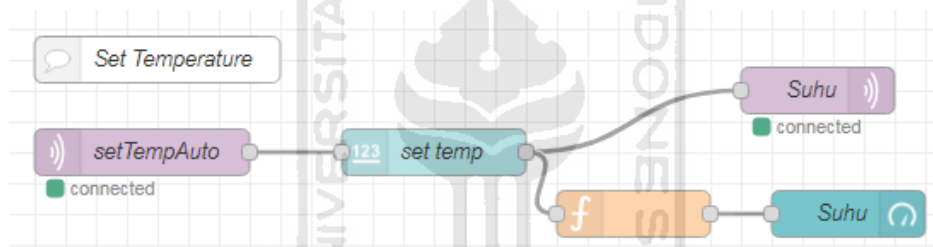
Gambar 3-13 menunjukkan contoh protokol-protokol yang digunakan dalam mengontrol AC merk Daikin yang sudah tersedia pada *library*. Berisi skrip untuk menyalakan dan mematikan AC, serta setting suhu dan kecepatan Fan AC. Dengan memodifikasi skrip pada library ini, dapat membantu pembuatan alat yang efisien sehingga tidak perlu mendapatkan data dari *remote control* AC yang asli.

3.6 Perancangan Flow Node pada NodeRED



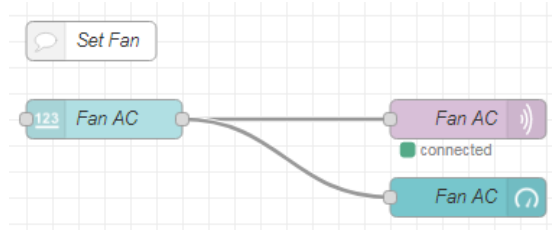
Gambar 3-15 Flow Node Switch On-Off

Gambar 3-9 menunjukkan flowchart dari node Switch pada NodeRED. MQTT in topik PIR mensubscribe topik PIR yang dipublish oleh ESP32. Node Switch menerima data 1 atau 0 baik secara otomatis dari MQTT maupun manual. Data kemudian dipublish oleh MQTT out topik Switch yang disubscribe oleh ESP32 untuk mengaktifkan dan menon-aktifkan AC.



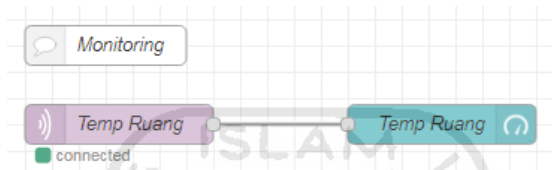
Gambar 3-16 Flow Node Set Suhu

Gambar 3-10 menunjukkan flowchart node set suhu. Set suhu bisa dilakukan secara otomatis maupun manual. NodeRED mensubscribe topik Set Temp Auto yang akan memberikan data berupa angka. Angka tadi akan masuk ke node numeric input yang bisa diset batas atas dan batas bawahnya (settingan suhu AC antar 18-28°C). Dari node numeric input data kan diteruskan ehingga menjadi tampilan berupa Bar pada dashboard antar muka NodeRED, dan juga akan dipublish dengan MQTT out topik Suhu.



Gambar 3-17 Flow Node Set Fan AC

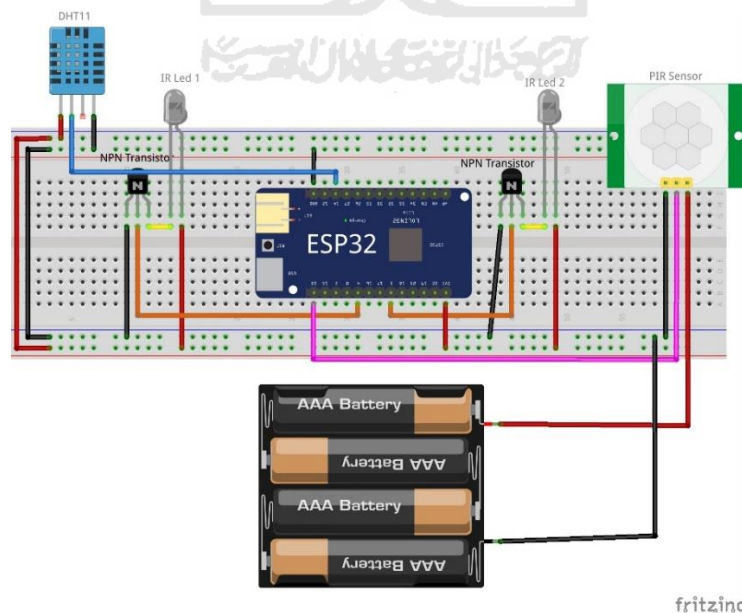
Set Fan AC mengatur kecepatan dari Fan AC. Fan AC ini hanya bisa diatur secara manual. Data dari NodeRED dipublish dengan MQTT out topik Fan AC. Data set Fan AC juga akan ditampilkan berupa Bar ada dashboard antar muka NodeRED.



Gambar 3-18 Flow Node monitoring ruangan

Flow yang terakhir ini hanya untuk menampilkan suhu ruangan pada dashboard antar muka NodeRED. Data didapatkan dari mensubscribe topik Temp Ruang. Kemudian data akan ditampilkan berupa Bar.

3.7 Perancangan rangkaian perangkat keras



Gambar 3-19 Desain rangkaian elektronik alat

Penjelasan komponen:

- a. DHT11 digunakan untuk memonitoring suhu ruangan. Penggunaannya mudah dan sudah berbentuk modul sehingga library sudah tersedia pada Arduino.
- b. IR Led menggunakan IR Led transmitter ukuran 5 mm.
- c. Transistor NPN 2N2222 digunakan untuk menambah daya jangkauan dari IR Led sesuai rekomendasi dari Arduino dalam pembuatan rangkaian IR Led dengan ESP32.
- d. PIR sensor yaitu HC-SR501 tidak menggunakan library pada Arduino sehingga mudah diaplikasikan.
- e. Daya menggunakan 4 buah baterai A2, dengan tegangan total 6V.

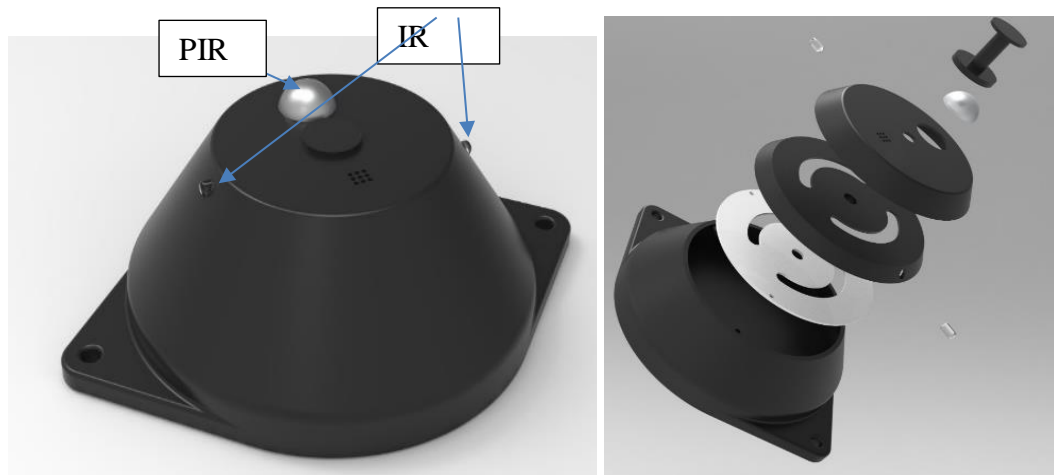
Keterangan rangkaian:

1. IR Led 1 dan Transistor NPN 2N2222 terhubung ke pin nomer 4 ESP32.
2. IR Led 2 dan Transistor NPN 2N2222 terhubung ke pin nomer 5 ESP32.
3. PIR sensor terhubung ke pin 13 ESP32.
4. DHT11 sensor terhubung ke pin 14 ESP32.
5. 4 baterai 1.5V (6V) digunakan sebagai daya input bagi PIR dan juga ESP32.



Gambar 3-20 Rangkaian elektronik alat

3.8 Desain 3D Alat



Gambar 3-21 Desain 3D alat

Desain 3D alat dilakukan pada software Solidworks. Prototype alat dibuat dengan metode 3D print dengan material PLA warna hitam. Alat ditempatkan pada langit-langit ruangan menggunakan baut. Sensor PIR akan menghadap bawah ke ruangan dan akan mendeteksi gerakan yang terjadi pada ruangan. Kedua IR Led bisa diputar menyesuaikan posisi AC yang digunakan.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Pembahasan

4.1.1 Pembahasan Program ESP32

```
#include "DHT.h"  
#include <WiFi.h>  
#include <PubSubClient.h>  
#include <Arduino.h>  
#include <IRremoteESP8266.h>  
#include <IRsend.h>  
#include <ir_Daikin.h>  
#include "EEPROM.h"
```

Program dibuka dengan memasukkan *library* yang akan digunakan. *Library* yang digunakan yaitu `#include "DHT.h"` untuk sensor DHT11, `#include <WiFi.h>` untuk modul wifi ESP32, `#include <PubSubClient.h>` untuk *publish* dan *subscribe* MQTT broker, `#include <IRremoteESP8266.h>` dan `#include <IRsend.h>` digunakan untuk *library* fungsi-fungsi remote AC yang ada, `#include <ir_Daikin.h>` untuk memilih protokol dan tipe AC yang digunakan. yang terakhir `#include "EEPROM.h"` digunakan untuk EEPROM internal pada ESP32.

```
const char* ssid = "Wergys";  
const char* password = "123456780";  
const char* mqtt_server = "test.mosquitto.org";  
  
WiFiClient espClient;  
PubSubClient client(espClient);
```

Skrip diatas digunakan untuk mendeklarasikan variabel-variabel wifi dan MQTT. Berisi ssid dan password untuk wifi dan juga mqtt server yang digunakan dari server MQTT broker Mosquitto.

```
int suhu;
int fan;
int pir = 14;
```

Skrip diatas untuk mendeklarasikan variabel suhu dan fan sebagai integer untuk setting AC. `int pir` digunakan untuk mendeklarasikan variabel sensor PIR pada pin 14 ESP32.

```
#define DHTPIN 13
#define DHTTYPE DHT11
DHT dht1(DHTPIN, DHTTYPE);
```

Skrip diatas untuk mendeklarasikan tipe sensor suhu DHT yaitu DHT11 pada pin 13. Dan juga variabel itu dimasukkan pada variabel instan `dht1`.

```
const uint16_t kIrLedD = 4;
IRDaikinESP ac1(kIrLedD);
IRsend irsend1(kIrLedD);

const uint16_t kIrLedP = 5;
IRDaikinESP ac2(kIrLedP);
IRsend irsend2(kIrLedP);
```

Skrip diatas untuk mendeklarasikan dua pin yang digunakan untuk 2 buah led inframerah yang digunakan. Pin yang digunakan pada ESP32 yaitu pin 4 dan pin 5. Kedua inframerah digunakan untuk mengontrol dua AC yang berbeda. Kedua tipe AC yang digunakan sama yaitu Daikin.

```
void setup_wifi() {

  delay(20);

  // We start by connecting to a WiFi network
  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());
}

```

Skrip diatas digunakan untuk mengatur proses koneksi wifi pada ESP32. Dengan mengambil ssid data password wifi yang telah dideklarasikan diatas untuk koneksi ke internet.

```

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      // ... and resubscribe
      client.subscribe("Switch");
      client.subscribe("Suhu");
      client.subscribe("FanAC");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
}
}

```

Void reconnect digunakan untuk mengkoneksikan ESP32 ke MQTT server. Ketika ESP32 sudah tersambung internet dan tersambung MQTT server maka ESP32 selanjutnya akan mensubscribe topik yaitu "Switch", "Suhu" dan "FanAC".

```

void setup() {
  dht1.begin();
  pinMode(pir, INPUT);
  ac1.begin();
  ac2.begin();
  irsend1.begin();
  irsend2.begin();
  EEPROM.begin(3);
  suhu = EEPROM.read(0);
  fan = EEPROM.read(1);
  Serial.begin(115200);

  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

```

Void setup pada ESP32 berisi pengaturan awal bagi ESP32. Skrip ini berisi untuk mengaktifkan DHT11, menjadikan sensor PIR menjadi input sensor, mengaktifkan ac1 dan ac2 sekaligus irsend1 dan irsend2. Mengaktifkan EEPROM internal yang akan digunakan untuk menyimpan variabel suhu dan fan AC. Variabel suhu dibaca dari EEPROM 0 dan variabel fan AC dibaca pada EEPROM 1. Serial dibuka pada data 115200. Yang terakhir adalah setup wifi dan MQTT.

```

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // nyalakan dan matikan AC
  if ((char)payload[0] == '1' && ((String)topic == "Switch")) {

    ACnyala();

  }
  else if ((char)payload[0] == '0' && ((String)topic == "Switch")) {

    ACmati();

  }
}

```



```

else if (((char)payload[0] >= '0') && ((char)payload[0] <= '9') && ((String)topic
== "Suhu")) {
    suhu = (char)payload[0];
    EEPROM.write(0, suhu);
    EEPROM.commit();
    ACnyala();
    ac1.send();
    ac2.send();

}

else if ((char)payload[0] >= '1' && (char)payload[0] <= '3' && ((String)topic ==
"FanAC")) {
    fan = (char)payload[0];
    EEPROM.write(1, fan);
    EEPROM.commit();
    ACnyala();
    ac1.send();
    ac2.send();
}
}
}

```

Skrip diatas termasuk pada void callback yang digunakan untuk mendapatkan data dari MWTT server untuk ESP32. Jadi ketika topik yang disubscribe oleh ESP32 mengirimka pesan maka pesan akan dibaca pada void callback ini. Dibagi dengan 3 program yang akan memilah 3 pesan dari 3 topik yaitu if untuk topik “Switch”, else if untuk topik “Suhu” dan else if untuk topik “FanAC”.

```

void ACnyala() {

    ac1.on();
    ac1.setFan(fan);
    ac1.setMode(kDaikinCool);
    ac1.setTemp(suhu);
    ac2.on();
    ac2.setFan(fan);
    ac2.setMode(kDaikinCool);
    ac2.setTemp(suhu);
    ac1.send();
    ac2.send();

}

void ACmati() {

```

```

ac1.setMode(kDaikinCool);
ac2.setMode(kDaikinCool);
ac1.off();
ac2.off();
ac1.send();
ac2.send();
}

```

Kedua skrip diatas berisi void yang digunakan pada void callback. Void AC nyala berisi protokol-protokol *remote control* untuk menyalakan AC. Void AC mati berisi protokol-protokol *remote control* untuk mematikan AC.

```

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  long now = millis();
  if (now - lastMsg > 1000) {
    lastMsg = now;

    // Read temperature as Celsius (the default)
    float t = dht1.readTemperature();
    // konversi nilai ke char array
    char tempString[8];
    dtostrf(t, 1, 2, tempString);
    Serial.print("Temperature: ");
    Serial.println(tempString);
    client.publish("roomTemp", tempString);
    if (t >= 30) {
      client.publish("setTempAuto", "3");
    }
    if (t <= 22) {
      client.publish("setTempAuto", "5");
    }
  }
  nilai = digitalRead(pir);

  if (nilai == HIGH) {

    if (pirStatus == LOW) {
      Serial.println("ada gerakan!");
      pirStatus = 1;
      client.publish("Jpir", "1");
    }
  }
}

```

```
} else {

  if (pirStatus == HIGH) {
    Serial.println("gerakan hilang!");
    pirStatus = 0;
    client.publish("Jpir", "0");
  }
}
}
```

Void loop disini berisi skrip untuk monitoring suhu ruangan yang langsung dipublish ke MQTT server dengan topik roomTemp. Pada skrip ini juga berisi dimana settingan suhu AC akan berubah secara otomatis, ketika suhu ruangan ≥ 30 derajat maka suhu akan diatur ke 21 derajat dan sebaliknya ketika suhu ruangan ≤ 22 maka suhu akan dinaikkan untuk menjadi 25 derajat. Pada skrip ini yang terakhir digunakan untuk membaca sensor PIR yang digunakan untuk mendeteksi gerakan. Ketika PIR mendeteksi gerakan maka ESP32 akan mem-publish pesan 1 pada topik "Jpir". Sebaliknya ketika PIR tidak mendeteksi gerakan maka ESP32 akan mem-publish pesan 0 dengan topik yang sama yaitu "Jpir".

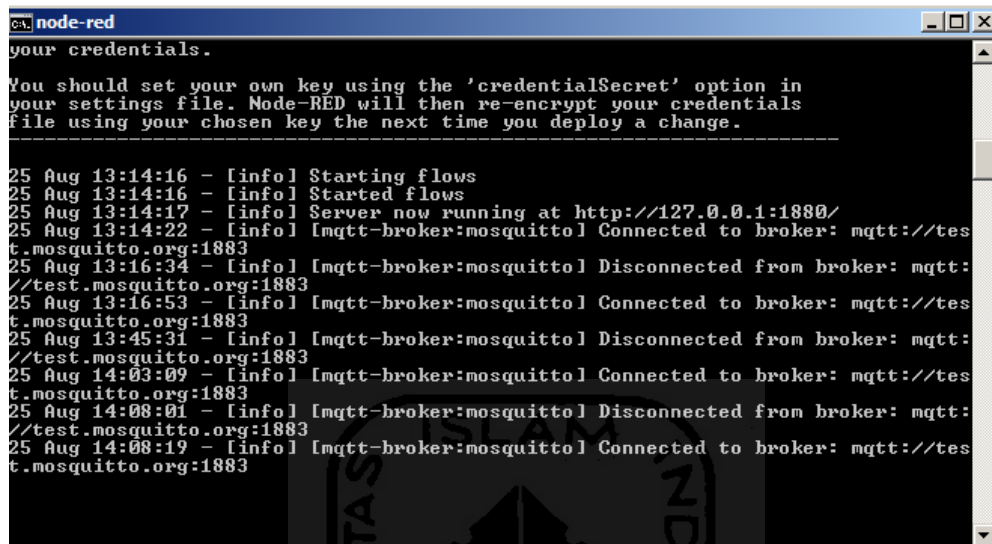


Gambar 4-1 Serial monitor ESP32

Terlihat pada serial monitor, bahwa suhu ruangan dimonitoring dan di-publish setiap detik. Terlihat juga pada serial monitor ketika ada gerakan, maka pir mengirim ke MQTT server dan dibaca oleh NodeRED yang kemudian mengirim pesan dengan topik Switch data 1 dan dibaca oleh ESP32 kemudian akan mengaktifkan AC.

4.1.2 Command Prompt Windows

Command prompt pada windows digunakan untuk mengaktifkan NodeRED dan juga MQTT. Semua aktifitas yang terjadi pada NodeRED terbaca pada command prompt.



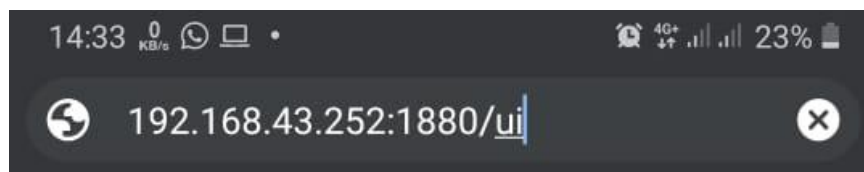
```
node-red
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

-----
25 Aug 13:14:16 - [info] Starting flows
25 Aug 13:14:16 - [info] Started flows
25 Aug 13:14:17 - [info] Server now running at http://127.0.0.1:1880/
25 Aug 13:14:22 - [info] [mqtt-broker:mosquitto] Connected to broker: mqtt://tes
t.mosquitto.org:1883
25 Aug 13:16:34 - [info] [mqtt-broker:mosquitto] Disconnected from broker: mqtt:
//test.mosquitto.org:1883
25 Aug 13:16:53 - [info] [mqtt-broker:mosquitto] Connected to broker: mqtt://tes
t.mosquitto.org:1883
25 Aug 13:45:31 - [info] [mqtt-broker:mosquitto] Disconnected from broker: mqtt:
//test.mosquitto.org:1883
25 Aug 14:03:09 - [info] [mqtt-broker:mosquitto] Connected to broker: mqtt://tes
t.mosquitto.org:1883
25 Aug 14:08:01 - [info] [mqtt-broker:mosquitto] Disconnected from broker: mqtt:
//test.mosquitto.org:1883
25 Aug 14:08:19 - [info] [mqtt-broker:mosquitto] Connected to broker: mqtt://tes
t.mosquitto.org:1883
```

Gambar 4-2 Comand prompt aktifitas NodeRED

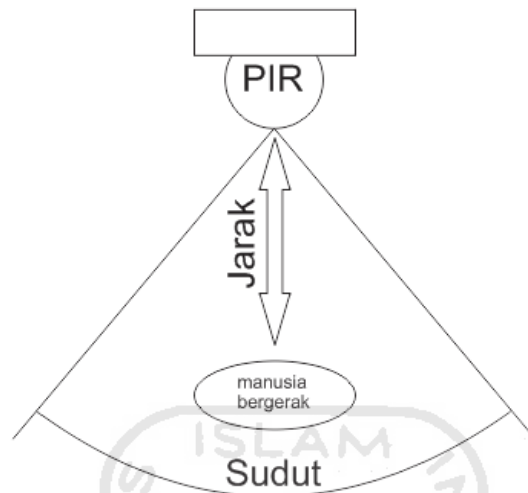
Terlihat pada gambar 4-2 diatas bahwa NodeRED aktif pada server <http://127.0.0.1:1880/> yang dapat dibuka langsung pada browser PC yang digunakan untuk mengaktifkan NodeRED. Sedangkan untuk membuka NodeRED yang sudah aktif pada PC atau gawai lain maka menggunakan IP address dari PC tersebut yang ditambahkan port untuk NodeRED yaitu :1880. MQTT broker juga sudah tersambung pada server mqtt://test.mosquitto.org:1883.



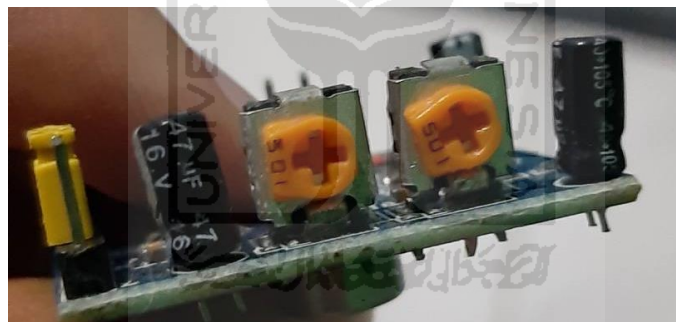
Gambar 4-3 Contoh IP Adress NodeRED pada browser *smartphone*

4.2 Hasil Pengujian

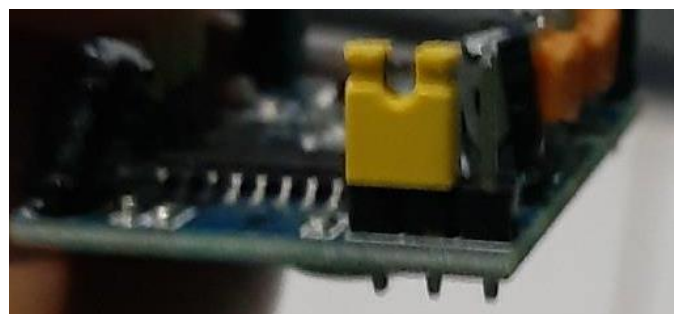
4.2.1 Hasil Pengujian Sensor PIR



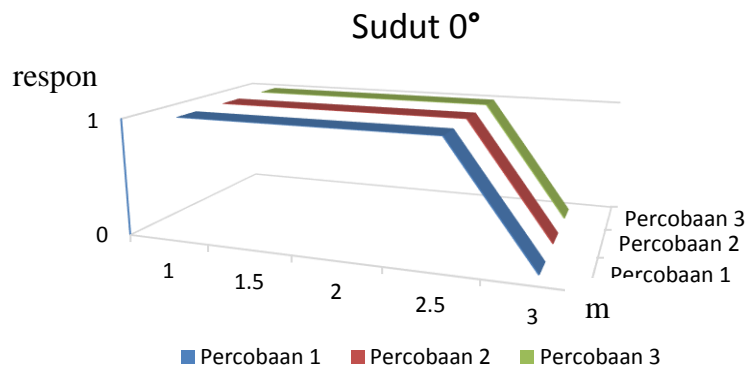
Gambar 4-4 Pandangan atas posisi pengujian



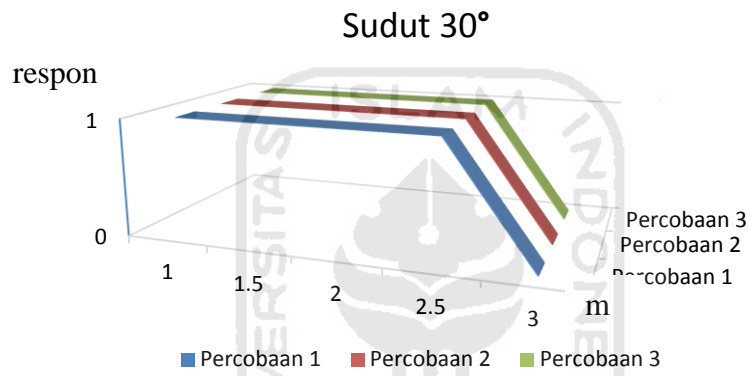
Gambar 4-5 Posisi regulator full



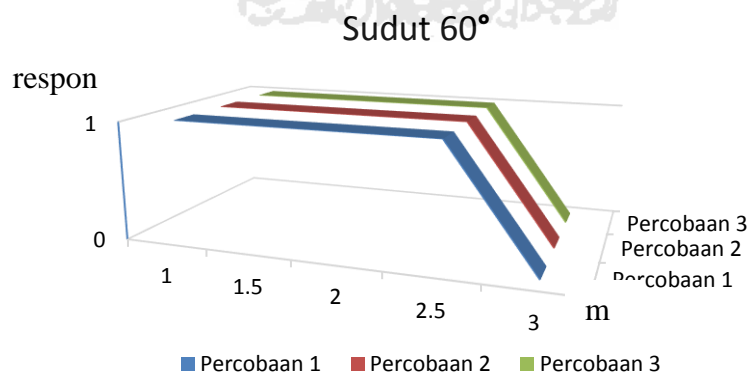
Gambar 4-6 Posisi jumper pada H



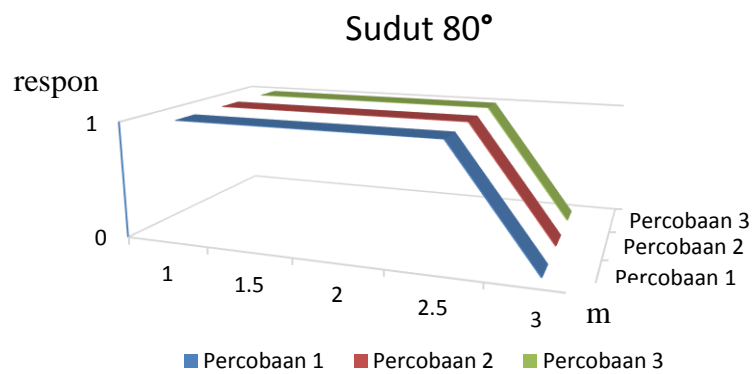
Gambar 4-7 Grafik pengujian sensor PIR Sudut 0°



Gambar 4-8 Grafik pengujian sensor PIR Sudut 30°



Gambar 4-9 Grafik pengujian sensor PIR Sudut 60°

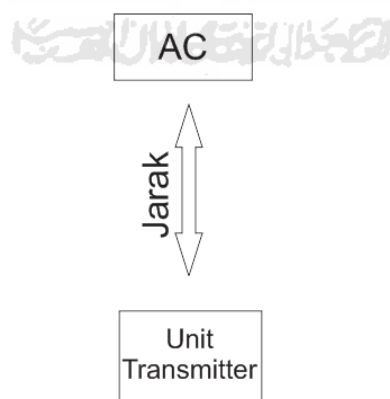


Gambar 4-10 Grafik pengujian sensor PIR Sudut 80°

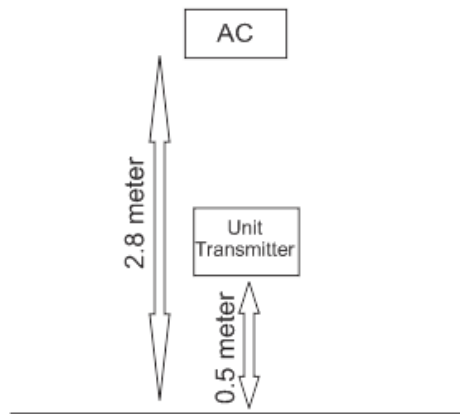
Dari hasil pengujian sensor PIR pada grafik gambar 4-4 hingga 4-7, terlihat bahwa sensor PIR hanya mampu aktif mendeteksi gerakan hingga pada jarak 2.5 meter dan dengan sudut kemiringan dari sensor PIR 80°. Dengan hasil berikut, sensor PIR sudah mampu untuk digunakan mendeteksi gerakan pada ruangan. Tinggi rata-rata ruangan berkisar 2.8 hingga 3 meter.

4.2.2 Hasil Pengujian IR Led Transmitter

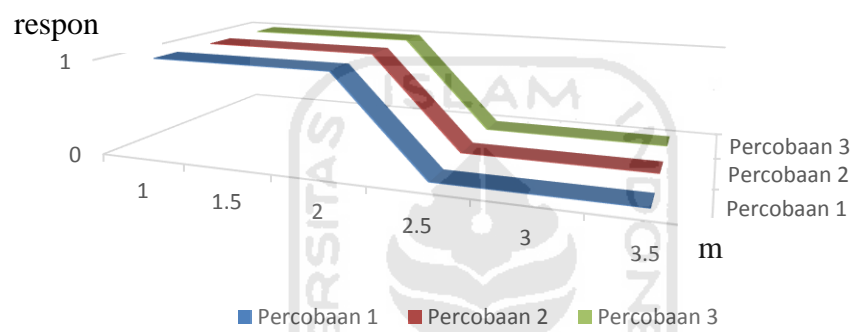
4.2.2.1 Posisi sejajar AC



Gambar 4-11 Pandangan atas posisi alat

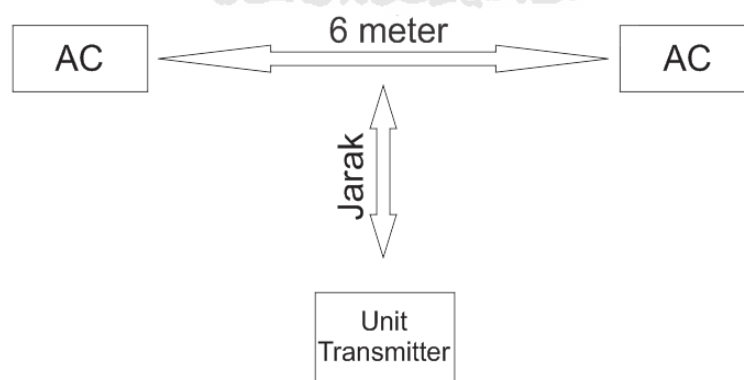


Gambar 4-12 Pandangan depan posisi alat

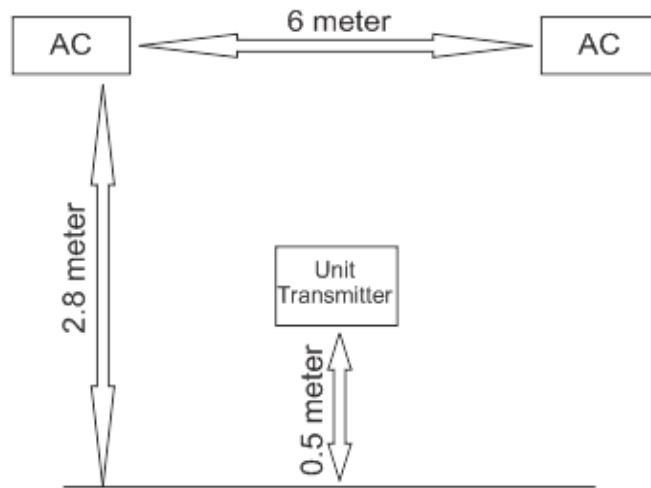


Gambar 4-13 Hasil pengujian IR Led Sejajar AC

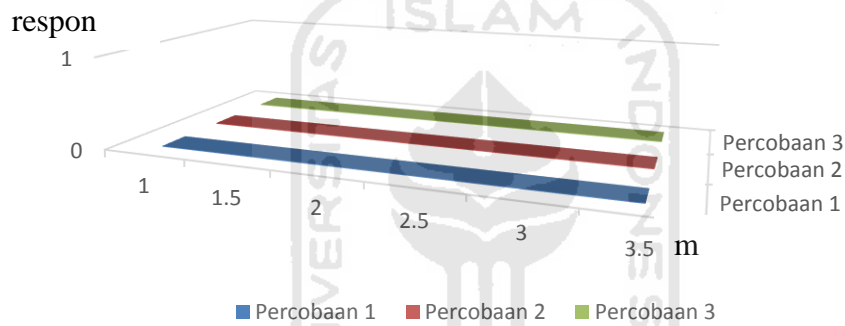
4.2.2.2 Posisi ditengah-tengah 2 AC



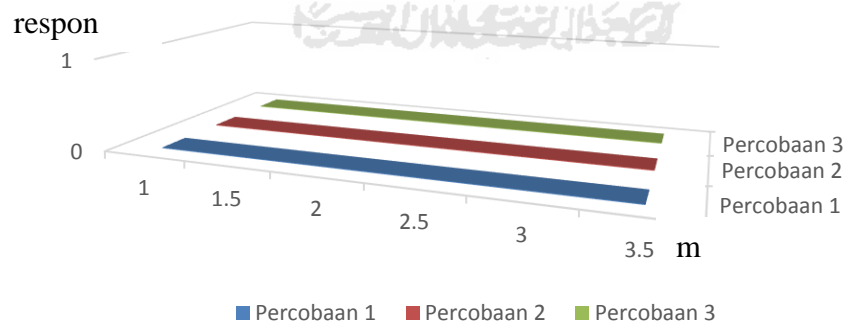
Gambar 4-14 Pandangan atas posisi alat



Gambar 4-15 Pandangan depan posisi alat



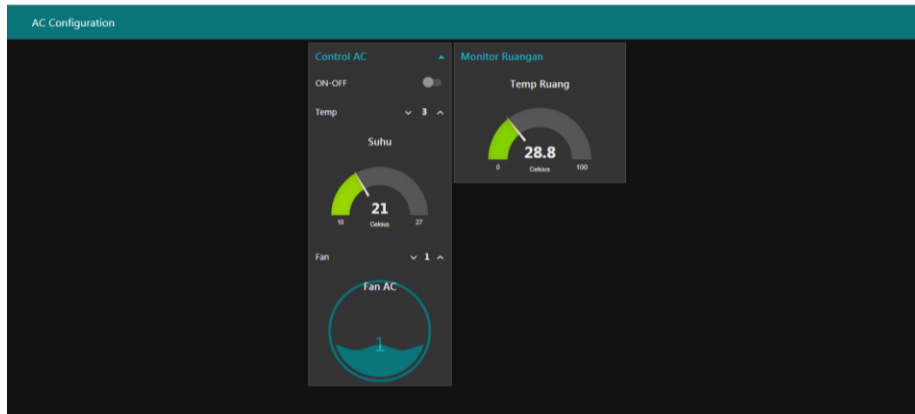
Gambar 4-16 Hasil pengujian IR Led 2 AC pada AC 1



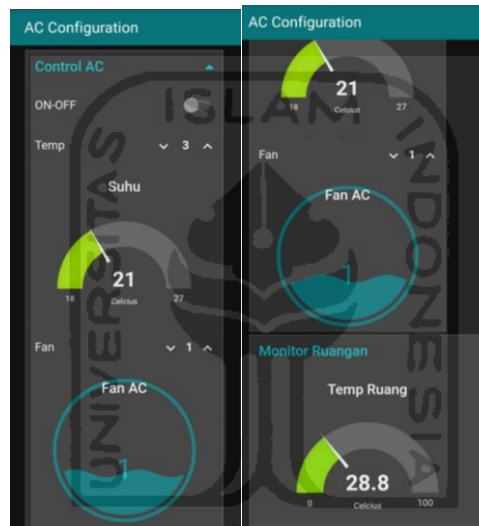
Gambar 4-17 Hasil pengujian IR Led 2 AC pada AC 2

Dari gambar 4-15 dan 4-16, hasil pengujian IR Led transmitter , kedua AC tidak merespon. Bahkan ketika jarak 1 meter dan ditengah-tengah dari kedua AC. Namun dari gambar 4-12, ketika posisi alat dan AC sejajar, AC mampu merespon hingga jarak 2 meter.

4.2.3 Hasil Pengujian Perangkat Lunak



Gambar 4-18 Dashboard NodeRED tampilan browser PC/Laptop



Gambar 4-19 Dashboard NodeRED tampilan browser smartphone

Tabel 4-1 Hasil pengujian perangkat lunak NodeRED

	Manual	Otomatis	Berfungsi
Switch On-Off	v	v	Ya
Set Suhu	v	v	Ya
Fan AC	v	-	Ya
Monitoring Ruangan	-	v	Ya

NodeRED sudah berfungsi sesuai yang diinginkan. Dashboard bisa diakses oleh setiap pengguna dengan memasukkan ip address yang digunakan oleh

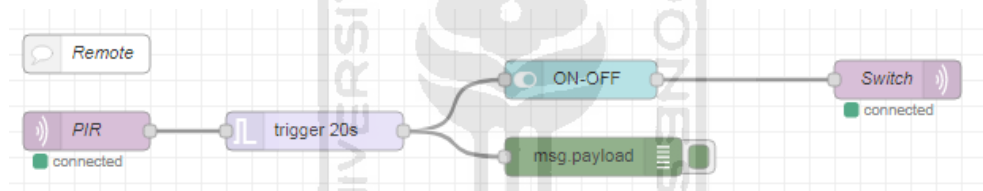
NodeRED. Jaringan yang digunakan hanya bisa secara lokal, sehingga pengguna harus berada pada satu koneksi jaringan wifi yang sama dengan NodeRED.

Suhu ruangan sudah mampu terbaca pada dashboard antar muka NodeRED. Setingan suhu dan Fan AC secara otomatis dan manual sudah bisa dilakukan. Kendala terjadi pada sensor PIR yang mengirimkan perintah 1 dan 0 terus menerus yang mengakibatkan AC hidup dan mati terus menerus ketika ruangan masih ditemukan gerakan atau orang.

4.3 Perbaikan

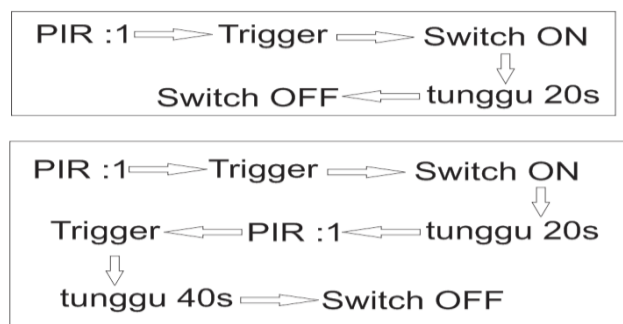
Terdapat dua kendala selama pengujian awal. Yaitu pada sensor PIR yang terus menerus mengirimkan sinyal ke NodeRED, dan juga AC belum merespon ketika IR Led digunakan untuk mengirim sinyal untuk 2 AC.

4.3.1 Perbaikan flow node Switch



Gambar 4-20 Penambahan node trigger

Penambahan node trigger ini digunakan untuk menyaring sinyal dari sensor PIR sehingga ketika PIR mengirim sinyal “1”, maka AC akan aktif. Namun off AC berasal dari timer node trigger yang sudah distell yaitu 20 detik. 20 detik ini dimulai ketika sensor PIR sudah tidak mengirimkan sinyal “1” lagi. Bisa dilihat alur dari gambar dibawah fungsi dari node trigger.

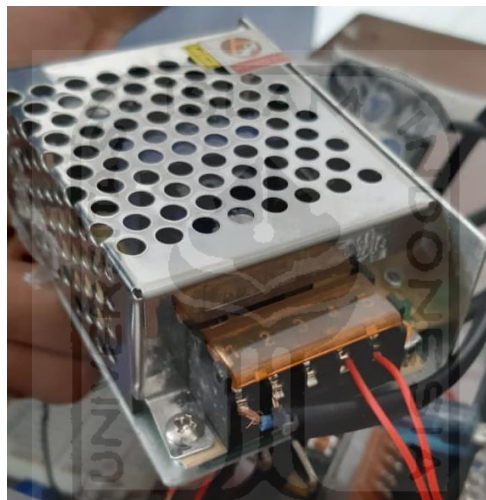


Gambar 4-21 Penjelasan node trigger

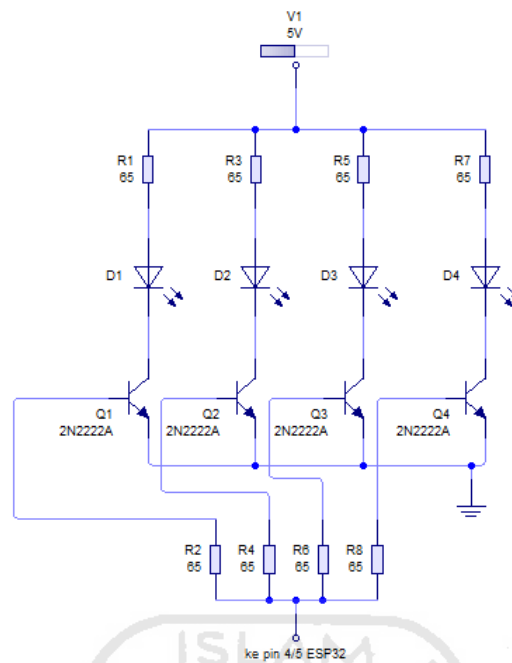
4.3.2 Perbaikan IR Led Transmitter

Karena jarak dan posisi dari kedua AC yang cukup jauh, dan kurang terangnya IR Led dalam memberikan sinyal perintah mengakibatkan kedua AC tidak merespon. IR Led perlu diterangkan cahayanya agar mampu menjangkau kedua AC. Alternative yang digunakan yaitu menambah jumlah IR Led yang digunakan agar cahaya IR Led lebih terang.

Bertambahnya IR Led yang digunakan berakibat pada penambahan penggunaan daya. Batre A2 4 buah dirasa kurang dalam memenuhi daya yang dibutuhkan untuk menyalakan seluruh rangkaian. Daya dirubah menggunakan Power Supply Unit yang mampu memberi tegangan 5V dan arus 3A.

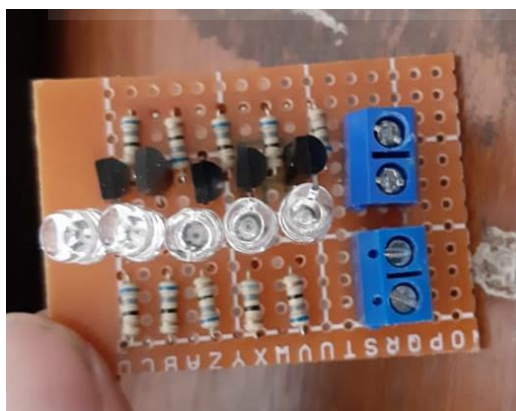


Gambar 4-22 PSU



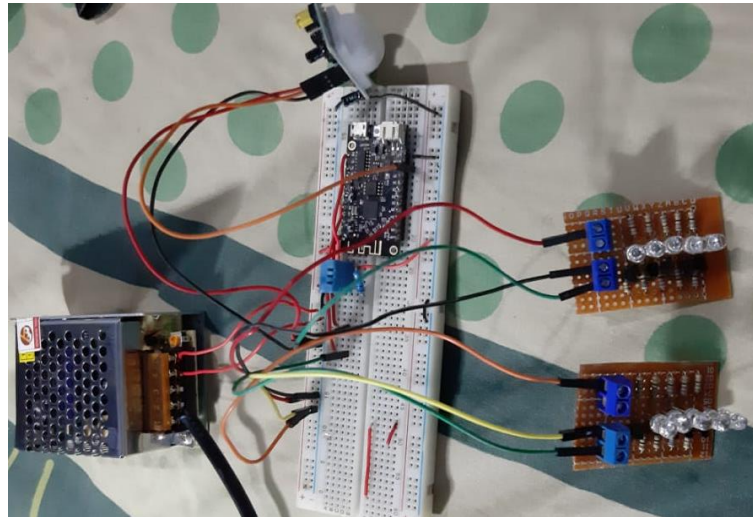
Gambar 4-23 Desain rangkaian IR Led Trasmmitter

Resistor 65 ohm digunakan pada masing masing IR Led. Transistor 2N2222A digunakan lagi untuk masing-masing IR Led. Arus dari pin 4 dan 5 ESP32 masuk ke base transistor dengan melalui resistor 65 ohm. Seluruh ground dihubungkan menjadi satu. IR Led yang digunakan masih sama yaitu Ir Led transmitter bening dengan ukuran 5mm. semua komponen dirakit pada PCB lubang.



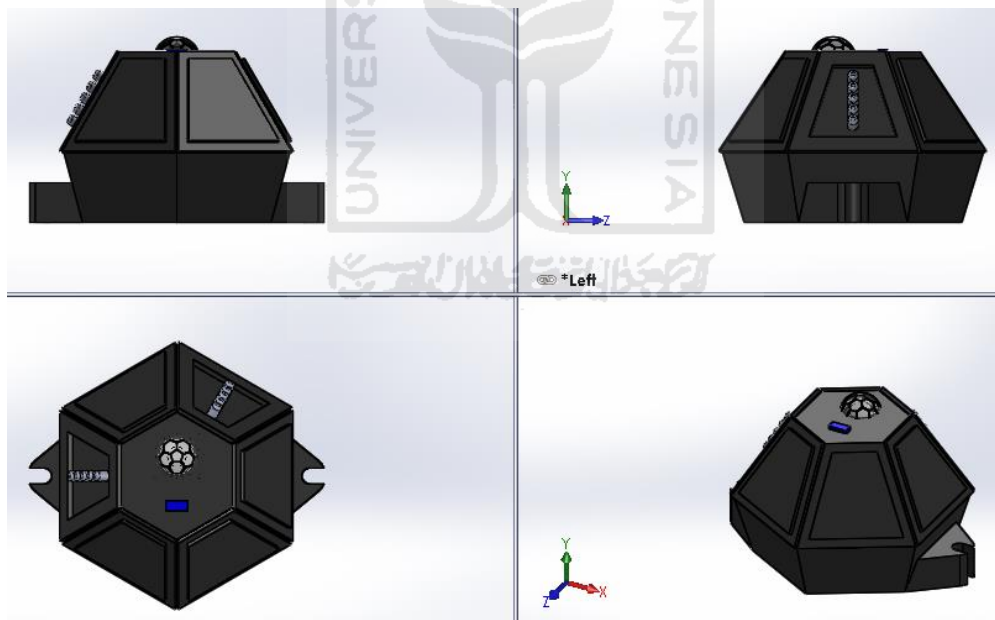
Gambar 4-24 Rangkaian IR Led Trasmmitter perbaikan

4.3.3 Rangkaian Elektronik Perbaikan



Gambar 4-25 Rangkaian elektronik alat perbaikan

4.3.4 Desain 3D perbaikan



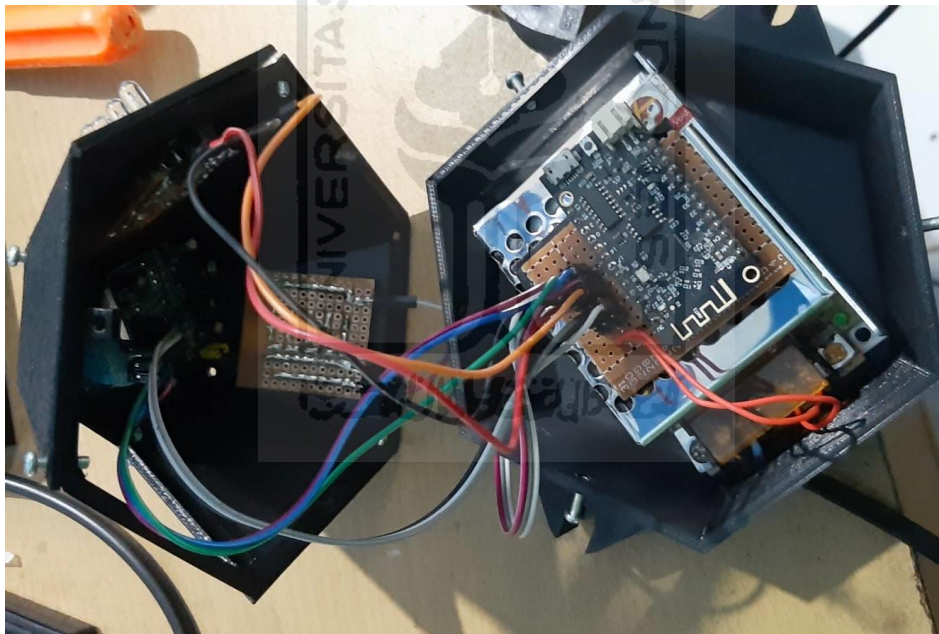
Gambar 4-26 Desain 3D perbaikan

Desain dirubah mengikuti bentuk IR Led transmitter. Memiliki 6 sisi, sehingga rangkaian IR Led Transmitter bisa menyesuaikan posisi AC pada ruangan. Rangkaian IR Led Transmitter bisa lebih dari 2. Selebihnya konsep masih sama seperti desain awal.

4.3.5 Hasil Perbaikan Prototype



Gambar 4-27 Prototipe



Gambar 4-28 Isi alat

PSU berada dibagian bawah dan dikunci dengan menggunakan lem pada *base cover*. PSU digunakan untuk mengaktifkan ESP32, sensor PIR dan kedua inframerah. Sedangkan untuk sensor DHT11 mengambil daya dari ESP32 yaitu 3.3V. ESP32 dipasang pada pin sisir yang sudah disoldir pada PCB bolong. Pin pada ESP32 ada 4 pin yang digunakan yaitu pin 4, 5, 13, 14, 3v dan ground. Sensor

PIR dan DHT11 terlihat diluar pada *upper case* sehingga mampu untuk mendeteksi gerakan dan mendeteksi suhu ruangan.

Desain dirubah menjadi segi 6 agar efisien dan mampu mengikuti posisi-posisi AC yang ada pada ruangan yang digunakan. desain terdiri dari bagian case bawah, case atas, dan cover sekaligus IR Led. Sambungan-sambungan yang ada menggunakan baut 3mm. Inframerah yang digunakan bisa ditambah tidak hanya 2 mengikuti jumlah AC yang ada pada ruangan. Alat nanti dipasang pada langit-langit ruangan yang akan dikontrol AC-nya.



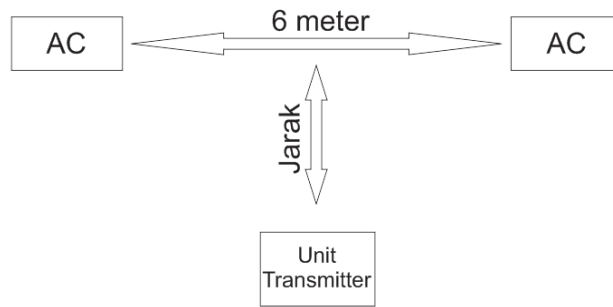
Gambar 4-29 Posisi kedua AC diruang CNC

4.4 Hasil Pengujian Perbaikan

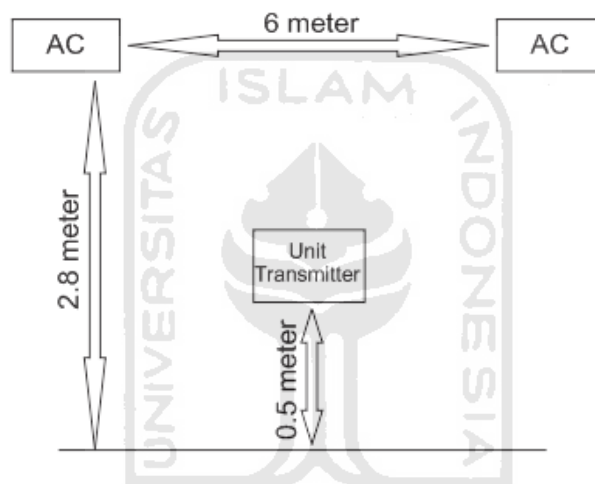
4.4.1 Hasil Pengujian flow node Switch

Node trigger bekerja sesuai yang diinginkan. AC tidak hidup-mati terus menerus, melainkan ketika masih ada orang diruangan atau masih terdeteksi gerakan maka AC tetap menyala dan akan nonaktif ketika sudah tidak ada gerakan dan ditambah delay waktu dari node trigger. Delay waktu pada node trigger dibuat 20 detik. 20 detik disini digunakan untuk mendeteksi apakah masih ada orang diruangan, dan juga mendeteksi ketika awal sensor PIR mengirim sinyal, apakah benar sinyal yang dikirim berupa gerakan yang terus-menerus atau bukan.

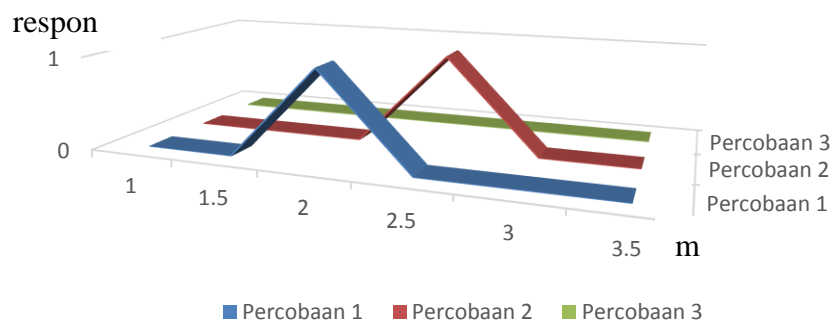
4.4.2 Hasil Pengujian Perbaikan IR Led Transmitter



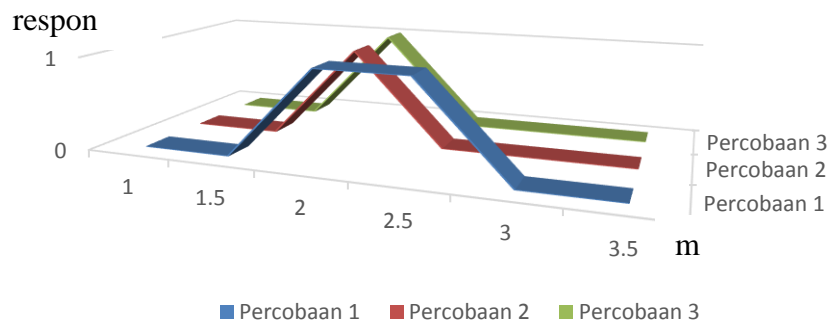
Gambar 4-30 Pandangan atas posisi alat



Gambar 4-31 Pandangan depan posisi alat



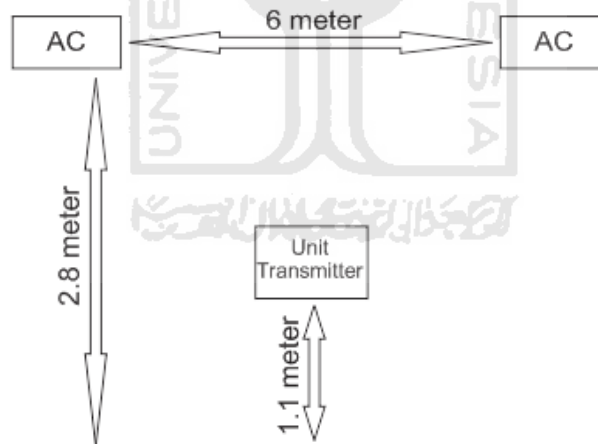
Gambar 4-32 Hasil perbaikan IR Led Transmitter 1 pada AC 1



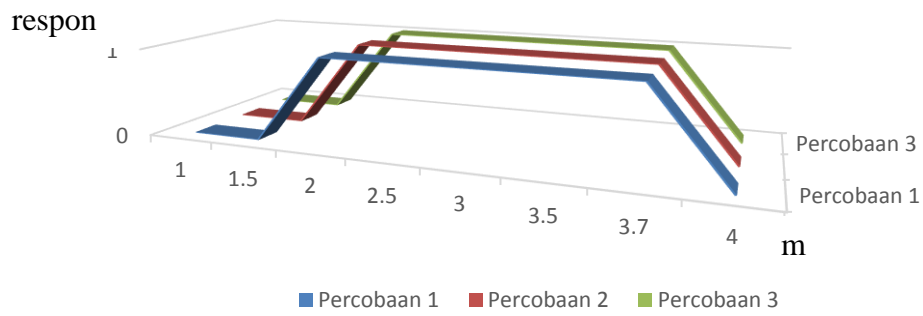
Gambar 4-33 Hasil perbaikan IR Led Transmitter 1 pada AC 2

Pada gambar 4-31 dan 4-32, ditemukan kondisi yaitu pada jarak 2-2.5 meter bahwa salah satu AC merespon dan AC lainnya tidak merespon. Dilakukan beberapa kali pengujian dan masih belum bisa kedua AC merespon sesuai yang diharapkan.

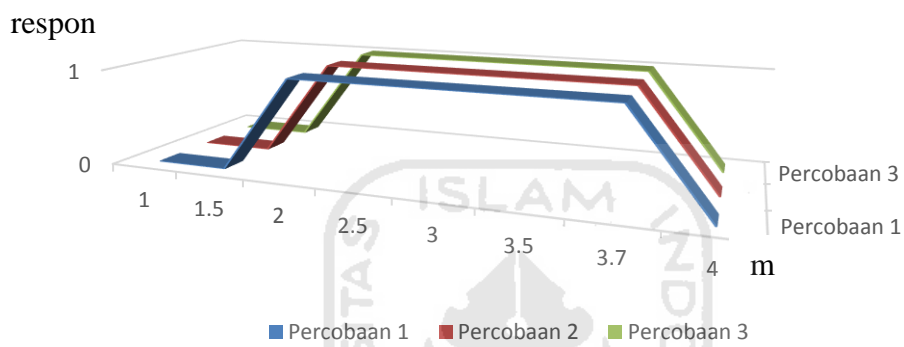
Dilakukan perubahan ketinggian pada posisi alat. Alat dinaikkan menjadi 1.1 meter dari tanah. Sehingga lebih sejajar kearah AC. Karena alat nantinya akan dipasang di langit-langit. Kondisi alat dipasang pada langit-langit sejajar dengan AC bila dilihat dari pandangan depan.



Gambar 4-34 Meningkatkan posisi alat



Gambar 4-35 Hasil perbaikan IR Led Transmitter 2 pada AC 1



Gambar 4-36 Hasil perbaikan IR Led Transmitter 2 pada AC 2

Terlihat pada hasil gambar 4-34 dan 4-35, kedua AC dapat merespon bersamaan hingga pada jarak 3.7 m. Pada jarak 4 meter AC sudah tidak merespon.

BAB 5

PENUTUP

5.1 Kesimpulan

- a) Alat monitoring dan kontrol yang dibuat mampu berjalan secara otomatis maupun manual menggunakan Internet of Things (IoT) dan protokol komunikasi MQTT dengan software webservice berbasis browser NodeRED.
- b) Sensor DHT11 mampu memonitoring suhu ruangan ber-AC.
- c) Sensor PIR yang digunakan mampu menjangkau hingga 2.5 meter dengan sudut 80°.
- d) IR Led yang sudah melalui tahap perbaikan mampu mengirim sinyal perintah ke dua buah AC pada ruangan dan kedua AC mampu merespon.
- e) Alat mampu dikontrol dan diakses oleh pengguna menggunakan PC/Laptop dan juga smartphone yang terkoneksi internet dengan memasukkan ip address lokal NodeRED.

5.2 Saran Perbaikan

- a) Dapat ditambahkan program untuk memilih merk AC yang akan dikontrol pada ruangan.
- b) Pembuatan rangkaian elektronik menggunakan pcb yang baik bukan hanya pcb bolong dan breadboard.
- c) Dashboard nantinya bisa ditambahkan untuk kontrol bermacam ruangan.
- d) Koneksi wifi untuk IP address NodeRED didaftarkan agar ketika terjadi reconnect IP address NodeRED tidak berubah.
- e) Koneksi wifi yang digunakan masih secara lokal, sehingga hanya bisa digunakan oleh pengguna dengan jaringan yang sama. Dibuat sehingga jaringan bisa luas tidak hanya lokal.

DAFTAR PUSTAKA

- Adyawadhana, Y. D. B. (n.d.). *TA : Integrasi Sistem Transaksi pada Vending Machine Menggunakan Cloud MQTT - Repositori Universitas Dinamika*. Retrieved October 12, 2020, from <http://repository.dinamika.ac.id/id/eprint/5207/>
- Ahmadi, S. (2019). Chapter 6—Internet of Things (NB-IoT and Massive MTC). In S. Ahmadi (Ed.), *5G NR* (pp. 747–787). Academic Press. <https://doi.org/10.1016/B978-0-08-102267-2.00006-3>
- Alvan Prastoyo Utomo, M., Aziz, A., Winarno, & Harjito, B. (2019). Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT). *Journal of Physics: Conference Series*, 1306, 012030. <https://doi.org/10.1088/1742-6596/1306/1/012030>
- Andreas, Aldawira, C. R., Putra, H. W., Hanafiah, N., Surjarwo, S., & Wibisurya, A. (2019). Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*, 157, 673–682. <https://doi.org/10.1016/j.procs.2019.08.218>
- Diori, G., Rianjani, D. A., Maulana, G., Tamzil, T. Z., Manawan, M., & Sukandi, A. (2019). Sistem Otomatisasi dan Monitoring Perawatan Berkala AC (Air Conditioner) Berbasis Arduino yang Terintegrasi IoT (Internet of Things). *Seminar Nasional Teknik Mesin*, 9(1), 184–193.
- Mukhopadhyay, B., Srirangarajan, S., & Kar, S. (2018). Modeling the analog response of passive infrared sensor. *Sensors and Actuators A: Physical*, 279, 65–74. <https://doi.org/10.1016/j.sna.2018.05.002>

- Natsir, M., Rendra, D. B., & Anggara, A. D. Y. (2019). IMPLEMENTASI IOT UNTUK SISTEM KENDALI AC OTOMATIS PADA RUANG KELAS DI UNIVERSITAS SERANG RAYA. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 6(1). <https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/1128>
- Pahrurrozi, M. (2020). ADAPTIVE CLASSROOM BERBASIS IOT (INTERNET OF THINGS), MANAJEMEN PENGGUNAAN AIR CONDITIONER (AC) SECARA OTOMATIS. *Publikasi Tugas Akhir S-1 PSTI FT-UNRAM*. <http://begawe.unram.ac.id/index.php/ta/article/view/13>
- Periyaldi, P., Bramanto, A., & Wajiansyah, A. (2018). Implementasi Sistem Monitoring Suhu Ruang Server Satnetcom Berbasis Internet Of Things (Iot) Menggunakan Protokol Komunikasi Message Queue Telemetry Transport (Mqtt). *JTT (Jurnal Teknologi Terpadu)*, 6(1), 23–29. <https://doi.org/10.32487/jtt.v6i1.435>
- Pradana, A., & Nurfiana, N. (2019). Rancang Bangun Monitor Dan Kontrol Suhu Ruang Server Menggunakan Perangkat Mobile Berbasis Internet Of Things (IoT). *Seminar Nasional Riset Terapan*, 4, A93–A98. <https://doi.org/10.31961/snrt.v4i0.413>
- Rafly, M. A. (2020). PEMROGRAMAN DAN FLOW DESIGN UNTUK RANGKAIAN LAMPU LIGHT EMITTING DIODE (LED) BERBASIS RASPBERRY PI 3B MENGGUNAKAN NODE-RED PADA RANCANG BANGUN GREENHOUSE AUTOMATION SYSTEM (GAS) [Preprint]. Open Science Framework. <https://doi.org/10.31219/osf.io/t3rvf>

Soelistijorini, R. (2019). IMPLEMENTASI ESP32 UNTUK MONITORING SUHU DAN KELEMBAPAN PADA FERMENTASI TEMPE BERBASIS IOT. *Prosiding Seminar Nasional Teknologi Elektro Terapan*, 1(1), 61–65.

NodeRed, diakses dari <https://nodered.org/> pada 30/12/2019



LAMPIRAN PROGRAM ESP32

```
#include "DHT.h"
#include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>
#include <ir_Daikin.h>
#include "EEPROM.h"

// Update these with values suitable for your network.
const char* ssid = "Wergys";
const char* password = "123456780";
const char* mqtt_server = "test.mosquitto.org";

//?

int suhu;
int fan;
int ledtrial = 15;

int pir = 14;

int nilai = 0;
int pirStatus = LOW;

//define dht pin
```



```

#define DHTPIN 13

//dht tipe
#define DHTTYPE DHT11

//set instan dht
DHT dht1(DHTPIN, DHTTYPE);

//setting ac dan IRled
const uint16_t kIrLedD = 4; // ESP8266 GPIO pin to use. Recommended: 4 (D2).
IRDaikinESP ac1(kIrLedD);
IRsend irsend1(kIrLedD);

const uint16_t kIrLedP = 5;
IRDaikinESP ac2(kIrLedP);
IRsend irsend2(kIrLedP);

//setting wifi dan PubSubclient
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup_wifi() {

  delay(20);

  // We start by connecting to a WiFi network
  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());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // nyalakan dan matikan AC
  if ((char)payload[0] == '1' && ((String)topic == "Switch")) {

```

```

ACnyala();

}
else if ((char)payload[0] == '0' && ((String)topic == "Switch")) {

ACmati();

}

else if (((char)payload[0] >= '0') && ((char)payload[0] <= '9') && ((String)topic
== "Suhu")) {
    suhu = (char)payload[0];
    EEPROM.write(0, suhu);
    EEPROM.commit();
    ACnyala();
    ac1.send();
    ac2.send();
}

else if ((char)payload[0] >= '1' && (char)payload[0] <= '3' && ((String)topic ==
"FanAC")) {
    fan = (char)payload[0];
    EEPROM.write(1, fan);
    EEPROM.commit();
    ACnyala();
    ac1.send();
    ac2.send();
}
}
}

```



```

void ACnyala() {

    ac1.on();
    ac1.setFan(fan);
    ac1.setMode(kDaikinCool);
    ac1.setTemp(suhu);
    ac2.on();
    ac2.setFan(fan);
    ac2.setMode(kDaikinCool);
    ac2.setTemp(suhu);
    ac1.send();
    ac2.send();

}

```

```

void ACmati() {
    ac1.setMode(kDaikinCool);
    ac2.setMode(kDaikinCool);
    ac1.off();
    ac2.off();
    ac1.send();
    ac2.send();
}

```

```

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID

```



```

String clientId = "ESP8266Client-";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str())) {
  Serial.println("connected");
  // ... and resubscribe
  client.subscribe("Switch");
  client.subscribe("Suhu");
  client.subscribe("FanAC");
} else {
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  // Wait 5 seconds before retrying
  delay(5000);
}
}
}

void setup() {
  dht1.begin();
  pinMode (ledtrial, OUTPUT);
  digitalWrite(ledtrial, LOW);

  pinMode(pir, INPUT);
  ac1.begin();
  ac2.begin();
  irsend1.begin();
  irsend2.begin();
  EEPROM.begin (3);
  suhu = EEPROM.read(0);

```



```

fan = EEPROM.read(1);
Serial.begin(115200);

setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);

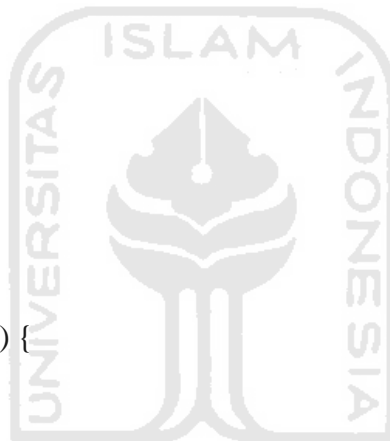
}

void loop() {

if (!client.connected()) {
  reconnect();
}
client.loop();
long now = millis();
if (now - lastMsg > 1000) {
  lastMsg = now;

  // Read temperature as Celsius (the default)
  float t = dht1.readTemperature();
  // konversi nilai ke char array
  char tempString[8];
  dtostrf(t, 1, 2, tempString);
  Serial.print("Temperature: ");
  Serial.println(tempString);
  client.publish("roomTemp", tempString);
  if (t >= 30) {
    client.publish("setTempAuto", "3");
  }
  if (t <= 22) {

```



```
    client.publish("setTempAuto", "5");  
  }  
}
```

```
nilai = digitalRead(pir);
```

```
if (nilai == HIGH) {
```

```
  if (pirStatus == LOW) {  
    Serial.println("ada gerakan!");  
    pirStatus = 1;  
    client.publish("Jpir", "1");  
  }
```

```
} else {
```

```
  if (pirStatus == HIGH) {  
    Serial.println("gerakan hilang!");  
    pirStatus = 0;  
    client.publish("Jpir", "0");  
  }
```

```
}  
  
}
```

