



**RANCANG BANGUN SISTEM MONITORING DAN KONTROL
SMART HOME MENGGUNAKAN PROTOKOL MQTT**

TUGAS AKHIR



Program Studi

S1 Teknik Komputer

Oleh :

**UNIVERSITAS
Dinamika**

Eka Fijianara Fernanda

17410200016

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

**RANCANG BANGUN SISTEM MONITORING DAN KONTROL SMART
HOME MENGGUNAKAN PROTOKOL MQTT**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Disusun Oleh :

Nama : Eka Fijianara Fernanda
NIM : 17410200016
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2022

TUGAS AKHIR

RANCANG BANGUN SISTEM MONITORING DAN KONTROL SMART HOME MENGGUNAKAN PROTOKOL MQTT

Dipersiapkan dan disusun oleh

Eka Fijianara Fernanda

NIM: 17410200016

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada: Senin, 17 Januari 2022

Susunan Dewan Pembahas

Pembimbing

I. Ira Puspasari, S.Si., M.T.

NIDN 0710078601

II. Harianto, S.Kom., M.Eng.

NIDN 0722087701

Pembahas :

I. Musayyanah, S.ST., M.T.

NIDN 0730069102


Digitally signed by Universitas
Dinamika
Date:
2022.01.17
21:53:37 +07'00'


Digitally signed by Universitas
Dinamika
Date:
2022.01.17
21:53:37 +07'00'


Digitally signed by Universitas
Dinamika
Date:
2022.01.17
21:53:37 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Digitally signed by
Universitas Dinamika
Date: 2022.01.26
08:46:17 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

“Do what you wanna do and enjoy it, because we only live once”



- Eka Fijianara F

UNIVERSITAS
Dinamika



*Karya ini saya persembahkan kepada Kedua Orang Tua, Keluarga Besar dan
Teman – Teman yang selalu memberikan Motivasi dan Dukungan.*

UNIVERSITAS
Dinamika

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Eka Fijianara Fernanda
NIM : 17.41020.0016
Program Studi : SI Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **RANCANG BANGUN SISTEM MONITORING DAN KONTROL
SMART HOME MENGGUNAKAN PROTOKOL MQTT**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta.
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditentukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 12 Januari 2022
Yang menyatakan



Eka Fijianara Fernanda
NIM : 17.41020.0016

ABSTRAK

Rumah merupakan kebutuhan yang sangat penting sebagai tempat berteduh. Seiring dengan berkembangnya jaman dan teknologi saat ini, semakin banyak sistem yang diterapkan di dalam perangkat-perangkat dirumah atau di dalam sebuah bangunan. *Smart home* merupakan sistem yang telah diprogram dan dapat bekerja dengan bantuan komputer untuk mengintegrasikan dan mengendalikan sebuah perangkat atau peralatan rumah secara otomatis dan efisien.

Pada penelitian ini bertujuan membuat rancang bangun monitoring dan kontrol *Smart home* menggunakan protokol MQTT. Pada rancangan tersebut, menggunakan 3 buah NodeMCU dengan 2 Node sebagai *Client* dan 1 sebagai *Server*. *Client* mengirimkan data hasil deteksi dari sensor ke *Server* dan diteruskan ke *Broker* sebagai penghubung *Publisher* dan *Subscriber*. Protokol MQTT sebagai media pengiriman pesan data dari *Server* ke *Smartphone* atau sebaliknya. Pesan yang dikirim dengan *smartphone* menggunakan data “0” dan “1” untuk sistem kontrol dan monitoring.

Hasil *delay* pengujian protokol MQTT diperoleh dari seluruh data yang dikirim dari *Server* ke *Broker* yaitu rata – rata sebesar 0.098s. Sehingga nilai *delay* yang dihasilkan masuk ke dalam kategori sangat bagus, dikarenakan memiliki nilai $< 0.15s$.

Kata Kunci : MQTT, *Broker*, *Smart home*, NodeMCU

KATA PENGANTAR

Bismillahirrahmannirrahiim. Puji dan syukur atas ke Hadirat Allah SWT, berkat Rahmat dan izin-Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “RANCANG BANGUN SISTEM MONITORING DAN KONTROL SMART HOME MENGGUNAKAN PROTOKOL MQTT”.

Dalam menyelesaikan Tugas Akhir ini penulis banyak mendapatkan bimbingan, dukungan dan motivasi dari berbagai pihak. Oleh karena itu penulis mengucapkan terimakasih banyak kepada:

1. Orang tua dan saudara – saudara penulis yang selalu memberikan doa, dukungan dan motivasi sehingga dapat menyelesaikan laporan Tugas Akhir ini.
2. Ibu Musayyanah , S.ST., M.T., selaku Dosen Pembahas yang telah memberikan banyak saran, motivasi dan ilmu dalam pengerjaan Tugas Akhir ini untuk menjadi lebih baik.
3. Ibu Ira Puspasari, S.Si., M.T., selaku Dosen Pembimbing yang telah memberikan inspirasi dan masukan agar Tugas Akhir ini menjadi lebih baik dalam pembuatan laporan Tugas Akhir.
4. Bapak Harianto, S.Kom., M.Eng. selaku dosen pembimbing yang telah memberi dukungan penuh berupa motivasi dan saran bagi penulis selama pelaksanaan Tugas Akhir.
5. Kepada Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer atas ijinnya yang diberikan untuk mengerjakan Tugas Akhir ini.
6. Seluruh staf dosen yang telah membimbing dan mengajarkan ilmunya kepada penulis.
7. Terimakasih kepada seluruh rekan mahasiswa S1 Teknik Komputer dan juga teman – teman yang sudah memberikan saran dan motivasi kepada penulis yang dapat menginspirasi bagi penulis dalam menyelesaikan laporan Tugas Akhir.

Semoga Allah SWT memberikan rahmat-Nya kepada seluruh pihak yang membantu penulis dalam penyelesaian Tugas Akhir.

Penulis menyadari bahwa masih terdapat dalam laporan Tugas Akhir ini. Oleh karena itu penulis mengharapkan saran dan kritik agar laporan Tugas Akhir lebih baik dan semoga laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak dan dapat menjadi bahan untuk penelitian selanjutnya.

Surabaya, 17 Januari 2022

Eka Fijianara Fernanda



UNIVERSITAS
Dinamika

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI.....	4
2.1 NodeMCU ESP-8266.....	4
2.2 DI-Relay 1	4
2.3 Module Sensor LDR	5
2.4 Sensor <i>Fingerprint</i>	5
2.5 Sensor MQ - 2	6
2.6 Motor <i>Servo</i>	6
2.7 Buzzer	7
2.8 MQTT	8
2.9 Quality Of Service.....	9
BAB III METODOLOGI PENELITIAN	12
3.1 Model Perancangan.....	12
3.1.1. Client 1	12
3.1.2. Client 2	13
3.1.3. Server & Publisher	15
3.1.4. MQTT Broker	16
3.1.5. <i>Subscriber</i> 1	17
3.1.6. <i>Subscriber</i> 2	17

3.2 Perancangan Perangkat Keras	17
3.3 Perancangan <i>Prototype</i>	19
3.4.1. MQTT Panel	20
BAB IV HASIL DAN PEMBAHASAN	21
4.1 Pengujian Pengiriman dan Penerimaan Pesan ke <i>MQTT Broker</i> Dari <i>Server</i>	21
4.1.1. Tujuan	21
4.1.2. Perlengkapan Yang Dipakai	21
4.1.3. Cara Pengujian.....	21
4.1.4. Hasil Uji	22
4.2 Pengujian Aplikasi mobile.....	26
4.2.1 Tujuan	26
4.2.2 Peralatan Yang Dipakai	26
4.2.3 Cara Pengujian.....	26
4.2.4 Pengujian Monitoring.....	27
4.2.5 Pengujian Controlling	28
4.3 Pengujian Transmisi Data dengan Protokol MQTT	28
4.3.1 Tujuan	28
4.3.2 Peralatan Yang Dipakai	28
4.3.3 Cara Pengujian.....	28
4.3.4 Hasil Pengujian.....	29
4.4 Hasil Pembahasan	30
BAB V PENUTUP	31
5.1. KESIMPULAN	31
5.2. SARAN	31
DAFTAR PUSTAKA	32
LAMPIRAN	33

DAFTAR GAMBAR

Gambar 2.1 Node Mcu ESP8266	4
Gambar 2.2 DI-Relay 1 Channel	5
Gambar 2.3 Module Sensor LDR.....	5
Gambar 2.4 Sensor Fingerprint.....	6
Gambar 2.5 LCD 16X2	6
Gambar 2.6 Servo.....	7
Gambar 2.7 Buzzer.....	7
Gambar 2.8 Arsitektur MQTT.....	8
Gambar 3.1 Model perancangan sistem.....	12
Gambar 3.2 Blok Diagram <i>Client 1</i>	13
Gambar 3.3 <i>Flowchart Client 1</i>	13
Gambar 3.4 Blok Diagram <i>Client 2</i>	14
Gambar 3.5 <i>Flowchart Client 2</i>	14
Gambar 3.6 Blok Diagram <i>Server</i>	15
Gambar 3.7 <i>Flowchart Server</i>	15
Gambar 3.8 Alur pemrosesan data <i>server</i> ke MQTT Broker.....	16
Gambar 3.9 Skematik <i>Server</i>	17
Gambar 3.10 Skematik <i>Client 1</i>	18
Gambar 3.11 Skematik <i>Client 2</i>	18
Gambar 3.12 Desain Pengemasan <i>Prototype</i>	19
Gambar 3.13 Rangkaian alat di ruang tamu.....	19
Gambar 3.14 Rangkaian alat di kamar.....	19
Gambar 3.15 Rangkaian alat di dapur.....	20
Gambar 3.16 Daftar fitur MQTT Panel.....	20
Gambar 4.1 Koneksi <i>Server</i> ke <i>WiFi</i>	22
Gambar 4.2 Koneksi <i>Client 1</i> & <i>Client 2</i> ke <i>Server</i>	22
Gambar 4.3 Hasil Pengamatan Pada Serial Monitor <i>Server</i> dan MQTT <i>Explorer</i>	23
Gambar 4.4 Kode pesan dari <i>Server</i>	25
Gambar 4.5 Menghubungkan <i>Server</i> ke MQTT Broker	27

Gambar 4.6 Hasil Monitoring pada Aplikasi MQTT Panel	27
Gambar 4.7 Hasil Kontrol Lampu dengan Aplikasi MQTT Panel	28



UNIVERSITAS
Dinamika

DAFTAR TABEL

Tabel 3.1 Kategori Latensi Besar <i>Delay</i>	10
Tabel 3.2 Kategori Degradasi <i>Packet Loss</i>	11
Tabel 4.1 Hasil pengujian <i>Client 1</i> dan <i>Client 2</i> dengan <i>Server</i>	23
Tabel 4.2 Hasil Pengujian Pengiriman Pesan dari <i>Server</i> ke MQTT Broker	24
Tabel 4.3 Hasil Pengujian <i>delay Server</i> dengan MQTT Broker	29



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1 Source Code	33
Lampiran 2 Hasil Turnitin	45



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rumah merupakan kebutuhan yang sangat penting sebagai tempat berteduh. Seiring dengan berkembangnya jaman dan teknologi, saat ini semakin banyak sistem yang diterapkan di dalam perangkat yang berada di rumah atau di dalam sebuah bangunan. Teknologi ini menggunakan sistem kerja cerdas dan otomatis dalam konsep kerjanya untuk membantu penghuni rumah dalam melakukan sesuatu agar dikerjakan lebih mudah (Aditya, et al., 2015). Teknologi tersebut adalah rumah pintar (*smart home*) yang dapat memberikan kenyamanan, keamanan dan efisiensi bagi pengguna. *Smart home* menggunakan sebuah teknologi yang memungkinkan bisa dikendalikan jarak jauh dengan jaringan internet yaitu IoT.

IoT (*Internet of Things*) adalah teknologi di mana benda-benda di sekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan Internet. Jadi Internet of Things (IoT) adalah sebuah konsep dimana suatu objek memiliki kemampuan untuk mentransfer data melalui jaringan internet (Widodo, et al., 2020). Kemajuan teknologi ini sudah menjadi semakin terkenal dalam kalangan masyarakat luas, begitu juga dengan penelitian sebelumnya yang membuat *smart home* dengan ide yang bervariasi, contohnya adalah “Perancangan Sistem *Smart Home* Dengan Konsep Internet Of Things Hybrid Berbasis Protokol Message Queuing Telemetry Transport” oleh Yohanes Bowo Widodo, menjadi salah satu referensi. Konsep yang dipakai oleh Yohanes yaitu IoT (*Internet of Things*) sehingga dapat diakses oleh multi client dengan protokol *MQTT*. Kekurangan dari penelitian dari tersebut, yaitu mikrokontroler NodeMCU yang digunakan hanya satu yang menyebabkan I/O tidak bisa terlalu banyak (Widodo, et al., 2020).

MQTT merupakan protokol yang difungsikan dalam IoT untuk proses pengiriman data. *MQTT* terdiri dari *publisher* dan *Subscriber* berbasis protokol yang mengizinkan banyak perangkat untuk berkomunikasi satu sama lain melalui *wireless*. Sehingga target sasaran dari penelitian ini adalah membuat sebuah rancang

bangun *smart home* yang bisa dimonitoring dan kontrol menggunakan protokol MQTT. Kelebihan yang dimiliki oleh Tugas Akhir ini yaitu menggunakan 3 buah mikrokontroler agar bisa digunakan di setiap ruangan dengan salah satu mikrokontroler sebagai *server* dan lainnya (*client*) menghubungkan ke *server* melalui IP *server* supaya tidak menggunakan kabel terlalu banyak.

Sistem pada *smart home* ini dirancang menggunakan beberapa sensor diantaranya Sensor *Fingerprint*, Modul Sensor MQ-2, Modul Sensor LDR (*Light Dependert Resistor*), 3 buah Mikrokontroler Node MCU dan protokol MQTT.

1.2 Rumusan Masalah

Dari penjelasan latar belakang dapat dirumuskan permasalahan pada Tugas Akhir ini yaitu :

1. Bagaimana merancang sebuah sistem *smart home* dengan mengintegrasikan beberapa sensor dengan tiga buah kontroler ?
2. Bagaimana menerapkan protokol MQTT agar 3 buah kontroler saling berkomunikasi ?
3. Bagaimana mengintegrasikan sebuah aplikasi *smartphone* monitoring dan kontroling dengan rangkaian elektronik kontroler yang telah dibuat ?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Sistem komunikasi antar NodeMCU ESP8266 (*server*) dengan *cloud* Broker menggunakan protokol MQTT.
2. Media yang digunakan untuk sistem kontrol menyalakan lampu menggunakan *smartphone*.
3. Fungsi alat hanya bisa digunakan jika alat dan *smartphone* tersambung ke koneksi internet.

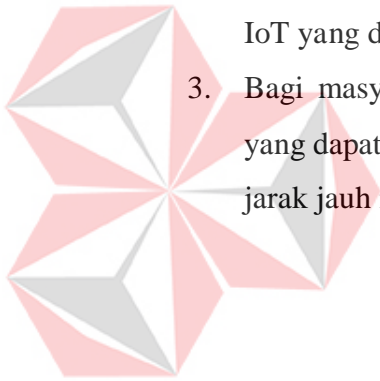
1.4 Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah di atas, dalam Tugas Akhir ini didapatkan tujuan pembuatan laporan sebagai berikut:

1. Merancang sebuah sistem *smart home* untuk mengintegrasikan beberapa sensor dengan tiga buah kontroler.
2. Menerapkan protokol MQTT agar 3 buah kontroler bisa saling berkomunikasi.
3. Mengintegrasikan sebuah aplikasi *smartphone* monitoring dan controlling dengan rangkaian elektronik kontroler yang telah dibuat.

1.5 Manfaat Penelitian

1. Bagi penulis: berpartisipasi terhadap kemajuan teknologi sebagai pengembangan mutu keamanan.
2. Bagi mahasiswa: memberikan pengetahuan tentang perkembangan teknologi IoT yang dapat dikombinasikan dengan MQTT.
3. Bagi masyarakat: memberikan pengetahuan tentang moderen teknologi IoT yang dapat di implementasikan pada *smart home* agar bisa dikendalikan secara jarak jauh menggunakan *smartphone*.



UNIVERSITAS
Dinamika

BAB II

LANDASAN TEORI

2.1 NodeMCU ESP-8266

NodeMCU pada dasarnya adalah pengembangan dari ESP8266 dengan firmware berbasis e-Lua, pada NodeMCU dilengkapi dengan micro usb port yang berfungsi untuk pemrograman maupun sebagai power supply untuk menyalakan NodeMCU. Selain itu pada NodeMCU dilengkapi dengan dua buah tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan package dari ESP8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan bahasa c hanya berbeda pada sintak. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploader (Ashari & Lidyawati, 2019). Gambar Node MCU ditunjukkan pada Gambar 2.1.



Gambar 2.1 Node Mcu ESP8266
(Sumber : www.embeddednesia.com)

2.2 DI-Relay 1

DI-Relay 1 adalah modul relay SPDT (*Single Pole Double Throw*) yang memiliki ketahanan yang lebih baik terhadap arus dan tegangan yang besar, baik dalam bentuk AC maupun DC. Sebagai *electronic switch* yang dapat digunakan untuk mengendalikan ON/OFF peralatan listrik berdaya besar (Tsauqi, et al., 2016). Gambar dari Relay dengan 1 channel ditunjukkan pada Gambar 2.2.

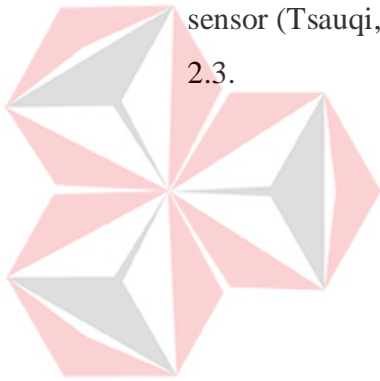


Gambar 2.2 DI-Relay 1 Channel

(Sumber: www.digitalapik.blogspot.com)

2.3 Module Sensor LDR

Module Sensor LDR adalah jenis modul yang berfungsi untuk mendeteksi intensitas cahaya dengan 2 *output* yaitu Digital dan Analog. Tegangan yang diperlukan untuk mengaktifkan modul sebesar 3.3V – 5V. Output digital berupa nilai 0 dan 1, sedangkan analog berupa nilai intensitas cahaya yang diterima oleh sensor (Tsauqi, et al., 2016). Gambar modul sensor LDR ditunjukkan pada Gambar 2.3.



Gambar 2.3 Module Sensor LDR

(Sumber: www.kelasrobot.com)

2.4 Sensor *Fingerprint*

Sensor Sidik Jari/*Fingerprint* merupakan sebuah alat elektronik yang berguna untuk memindai sidik jari seseorang dengan cara scanning identitas. Sensor Fingerprint sering dipakai pada beberapa peralatan elektronik seperti *smartphone*, pintu masuk rumah/kamar, alat absensi karyawan dan berbagai macam peralatan elektronik yang membutuhkan tingkat keamanan yang tinggi, dan hanya bisa di akses oleh pengguna itu sendiri (Arsyad, et al., 2021). Gambar sensor *fingerprint* ditunjukkan Pada Gambar 2.4.

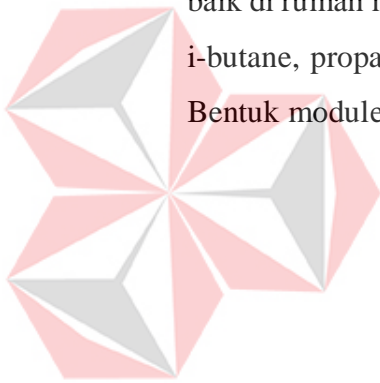


Gambar 2.4 Sensor Fingerprint

(Sumber: www.indomaker.com)

2.5 Sensor MQ - 2

Sensor MQ-2 adalah sensor yang digunakan untuk mendeteksi konsentrasi gas yang mudah terbakar di udara serta asap dan output membaca sebagai tegangan analog. Sensor gas asap MQ-2 dapat langsung diatur sensitivitasnya dengan memutar trimpot nya. Sensor ini biasa digunakan untuk mendeteksi kebocoran gas baik di rumah maupun di industri. Gas yang dapat dideteksi diantaranya: Gas LPG, i-butane, propane, methane, alcohol, Hydrogen, smoke (Sarmidi & Fauzi, 2019). Bentuk module sensor MQ2 ditunjukkan pada Gambar 2.5.



Gambar 2.5 LCD 16X2

(Sumber: otomasi.sv.ugm.ac.id)

2.6 Motor Servo

Motor Servo merupakan motor yang mampu bekerja secara dua arah, *motor servo* bekerja dengan sistem *closed feedback* dimana posisi dari *motor servo* akan diinformasikan kembali ke rangkaian kontrol yang ada didalam *motor servo*. *Motor servo* terdiri dari sebuah *motor*, rangkaian *gear*, *potensiometer*, serta rangkaian *control*. *Potensiometer* pada *motor servo* berfungsi sebagai penentu batas sudut dari putaran *servo*. *Motor servo* biasanya hanya bergerak mencapai sudut tertentu saja

dan tidak secara kontinyu (Rinaldy, et al., 2013). Bentuk dari motor servo ditunjukkan pada Gambar 2.6.



Gambar 2.6 Servo

(Sumber: www.sinaupedia.com)

2.7 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak balik sehingga membuat udara bergetar yang menghasilkan suara (Sarmidi & Fauzi, 2019). Bentuk dari komponen buzzer ditunjukkan pada Gambar 2.7.



Gambar 2.7 Buzzer

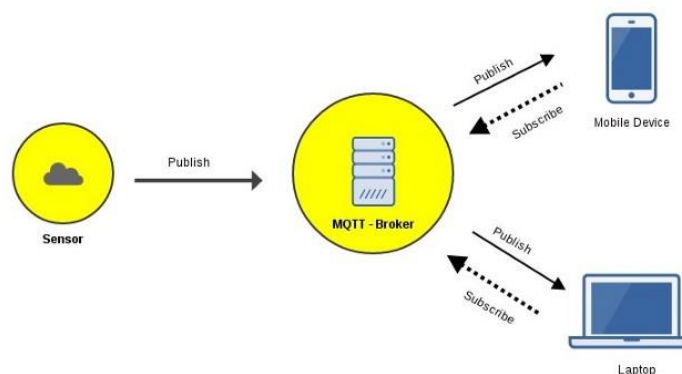
(Sumber: www.belajarmikrokontroler-2018.blogspot.com)

2.8 MQTT

Protokol MQTT (*Message Queue Telemetry Transport*) adalah protokol pesan ringan (*lightweight*) berbasis publish-subscribe digunakan di atas protokol TCP/IP. Protokol ini mempunyai ukuran paket data *low overhead* kecil (minimal 2 gigabyte) dengan konsumsi daya kecil. MQTT bersifat terbuka, simpel dan didesain agar mudah untuk diimplementasikan, yang mampu menangani ribuan *client* jarak jauh dengan hanya satu *server*. Broker bertanggung jawab untuk mendistribusikan pesan ke *client* tertarik berdasarkan topik pesan (Saputra, et al., 2017).

Berikut merupakan fitur protokol MQTT:

1. *Publish/subscribe message pattern* yang menyediakan distribusi *message* darisatu ke banyak dan *decoupling* aplikasi.
2. *Messaging transport* yang *agnostic* dengan isi dari payload.
3. Menggunakan TCP/IP sebagai konektivitas dasar jaringan.
4. Terdapat tiga level *Qualities of Service* (Qos) dalam penyampaian pesan :
 - 4.1 “*At most once*”, di mana pesan dikirim dengan upaya terbaik dari jaringan TCP/IP. Kehilangan pesan atau terjadi duplikasi dapat terjadi.
 - 4.2 “*At least once*”, dapat dipastikan pesan tersampaikan walaupun duplikasi dapat terjadi.
 - 4.3 “*Exactly once*”, dimana pesan dapat dipastikan tiba tepat satu kali.



Gambar 2.8 Arsitektur MQTT

(sumber : <https://medium.com>)

Sistem umum MQTT seperti Pada Gambar di atas membutuhkan dua komponent perangkat lunak utama yaitu :

- **MQTT Client** yang nantinya akan di install di device. Untuk Arduino anda bisa memakai **pubsubclient**, pustaka seperti **mqtt.js** bisa dipakai pada platform Node.js di Raspberry Pi ataupun laptop.
- **MQTT Broker** yang berfungsi untuk menangani publish dan subscribe data. Untuk platform Node.js anda bisa memakai broker **mosca** sedangkan untuk platform yg lain banyak broker tersedia seperti mosquitto, HiveMQ dll.

Keuntungan dari sistem **publish/subscribe** adalah antara sumber pengirim data (publisher) dan penerima data (*client*) tidak saling mengetahui karena ada broker diantara mereka atau istilah kerennya yaitu **space decoupling** dan yang lebih penting lagi yaitu adanya **time decoupling** dimana publisher dan *client* tidak perlu terkoneksi secara bersamaan, misalnya *client* bisa saja disconnect setelah melakukan subscribe ke broker dan beberapa saat kemudian *client* connect kembali ke broker dan *client* tetap akan menerima data yang terpending sebelumnya proses ini dikenal dengan mode **offline** (Pr., 2015).

2.9 Quality Of Service

Quality of Service (QoS) adalah suatu proses pengukuran tentang kualitas layanan jaringan. Berikut ini merupakan Parameter QoS:

1) Delay

Delay atau *Latency* adalah durasi waktu yang diperlukan dalam proses pengiriman data. Misalnya seorang kurir mengantarkan paket dari pos A ke pos F dengan waktu penerimaan sekian, selisih waktu antara paket diterima dengan waktu paket dikirim disebut sebagai *delay* atau *latency*. Kategori latensi besar *delay* ditunjukkan pada Tabel 3.1 dan dapat dirumuskan pada persamaan 3.1 sebagai berikut :

$$Delay = Tr - Ts....(3.1)$$

Keterangan :

Tr = Waktu terima data (detik)

Ts = Waktu kirim data (detik)

Tabel 3.1 Kategori Latensi Besar *Delay*

Kategori Latensi	Jumlah <i>Delay</i>
Sangat Bagus	< 0.15 s
Bagus	0.15 s s/d 0.3 s
Sedang	0.3 s s/d 0.45 s
Buruk	> 0.45 s

2) *Throughput*

Kecepatan (*rate*) transfer data dengan satuanbps. *Throughput* adalah sejumlah total penerimaan data yang berhasil di pantau pada *sebuah* destinasi selama kurun waktu tertentu. *Throughput* dapat dihitung dengan rumus :

$$\text{Throughput} = \frac{\text{Paket data yang diterima}}{\text{Lama pengamatan}}$$

3) *Packet Loss*

Packet loss adalah sekumpulan data yang menghilang setiap detiknya. *Packet loss* disebabkan oleh beberapa hal, signal internet yang buruk, antrian pengiriman data, berpapasan dengan paket data yang lain, dan kesalahan perangkat keras jaringan. Kategori *packet loss* ditunjukkan pada Tabel 3.2. *Packet loss* dapat dirumuskan pada persamaan 3.2 sabagai berikut :

$$\text{Packet loss} = (Pd / Ps) \times 100\%....(3.2)$$

Keterangan :

Pd = Total paket yang gagal (paket)

Ps = Total paket terkirim (paket)

Tabel 3.2 Kategori *Packet Loss*

Kategori Degradasi	Packet Loss
Sangat Bagus	0%
Bagus	3%
Jelek	15%
Sangat Jelek	25%

2.9 *Smart Home*

Smart home didefinisikan sebagai tempat tinggal yang dilengkapi dengan komputasi data dan teknologi informasi yang dapat merespon kebutuhan penghuni rumah, bekerja dengan mengandalkan efisiensi, kenyamanan, keamanan, penghematan, dan hiburan yang bisa didapatkan melalui manajemen teknologi dalam rumah dan koneksi ke dunia luar. Dalam operasinya, *smart home* dibantu oleh komputer untuk memberikan fasilitas fasilitas yang diinginkan secara otomatis dan sudah terprogram. Perintah dan sistem kendali *smart home* dapat dilakukan dengan suara, remote kontrol dengan kendali jarak jauh, tepukan tangan, sensor, dan sebagainya (Aditya, et al., 2015)

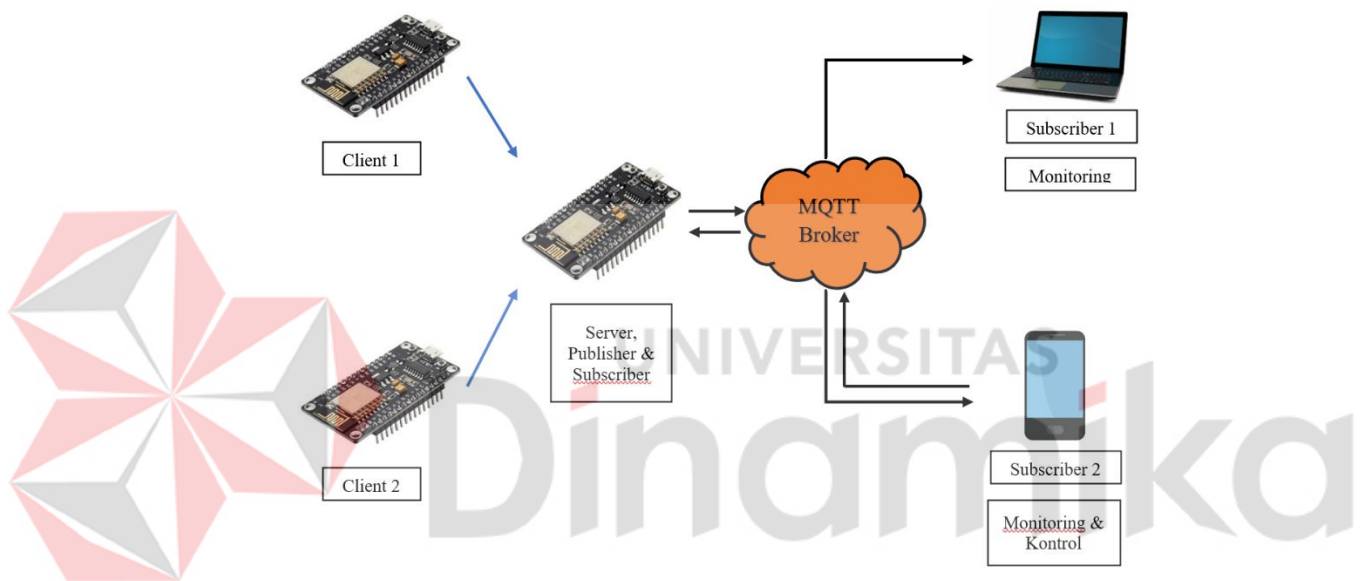


UNIVERSITAS
Dinamika

BAB III METODOLOGI PENELITIAN

3.1 Model Perancangan

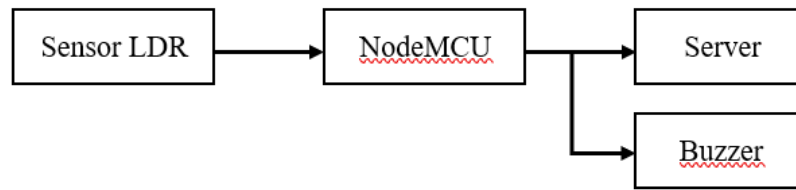
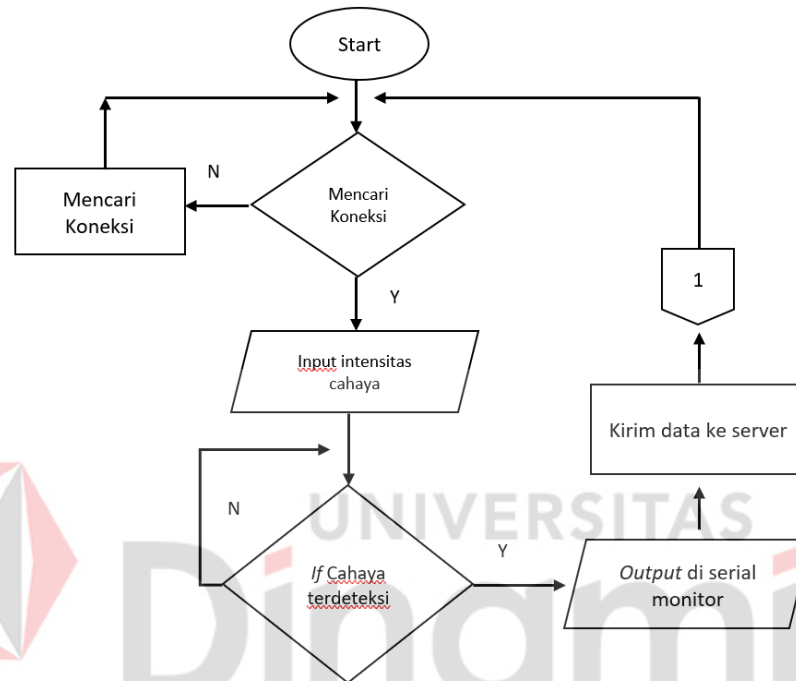
Pada Gambar 3.1 menunjukkan ada beberapa bagian dari model perancangan sistem yang memiliki fungsi tersendiri, berikut adalah daftar dan penjelasan dari setiap bagiannya.



Gambar 3.1 Model perancangan sistem

3.1.1. Client 1

Client 1 tugasnya yaitu menerima data dari sensor LDR yang didapat dari hasil mendeteksi intensitas cahaya dan mengirimkan data tersebut ke *Server* berupa sebuah pesan “*Client 0*” “Masuk” dan “*Client 1*” “No Data”. “*Client 0*” dan “*Client 1*” dibuat berbeda agar *Server* bisa membedakan asal dari pesan tersebut. *Flowchart* dan blok diagram dari *Client 1* ditunjukkan pada Gambar 3.2 dan 3.3.

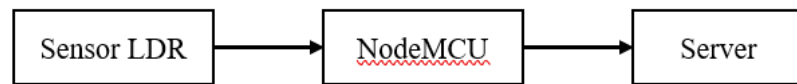
Gambar 3.2 Blok Diagram *Client 1*Gambar 3.3 Flowchart *Client 1*

Alur *flowchart Client 1* di atas diawali dengan proses menghubungkan ke koneksi *WiFi* lokal setelah itu menghubungkan *Client 1* ke *server* melalui *IPAddress* yang dimiliki oleh *server* untuk proses pengiriman datanya. Input dari *Client 1* adalah jumlah intensitas cahaya yang dideteksi oleh sensor LDR. Apabila terdeteksi jumlah intensitas cahaya dengan batas yang telah ditentukan yaitu bernilai 512, maka *Client 1* akan mengirimkan sebuah pesan ke *Server* dan juga nilai intensitas cahaya tersebut bisa dipantau di serial monitor.

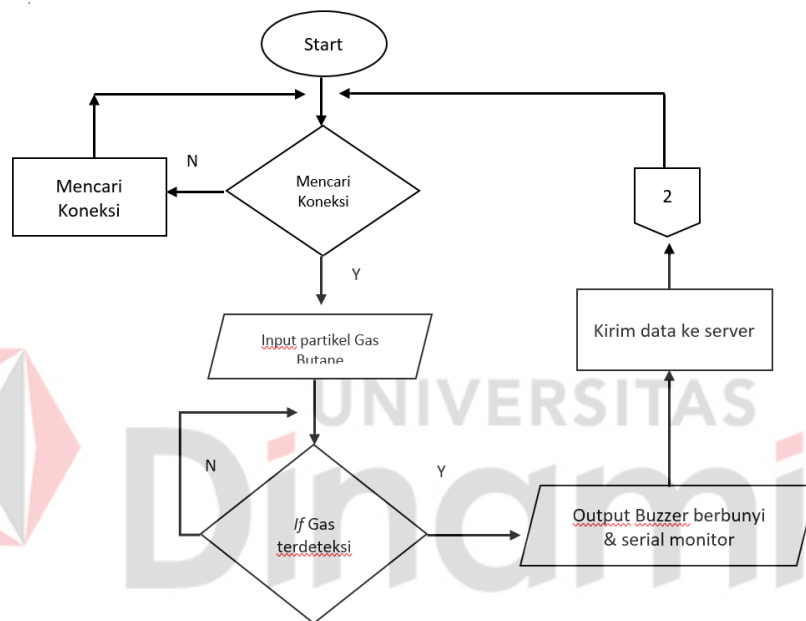
3.1.2. Client 2

Client 2 tugasnya yaitu menerima data dari sensor MQ2 yang didapat dari hasil mendeteksi jumlah partikel asap dan mengirimkan data tersebut ke *server*

berupa sebuah pesan “*Client 2*” “Masuk” dan “*Client 3*” “No Data”. “*Client 2*” dan “*Client 3*” dibuat berbeda agar *Server* bisa membedakan asal dari pesan tersebut. *Flowchart* dan blok diagram dari *Client 1* ditunjukkan pada Gambar 3.4 dan 3.5.



Gambar 3.4 Blok Diagram *Client 2*



Gambar 3.5 *Flowchart Client 2*

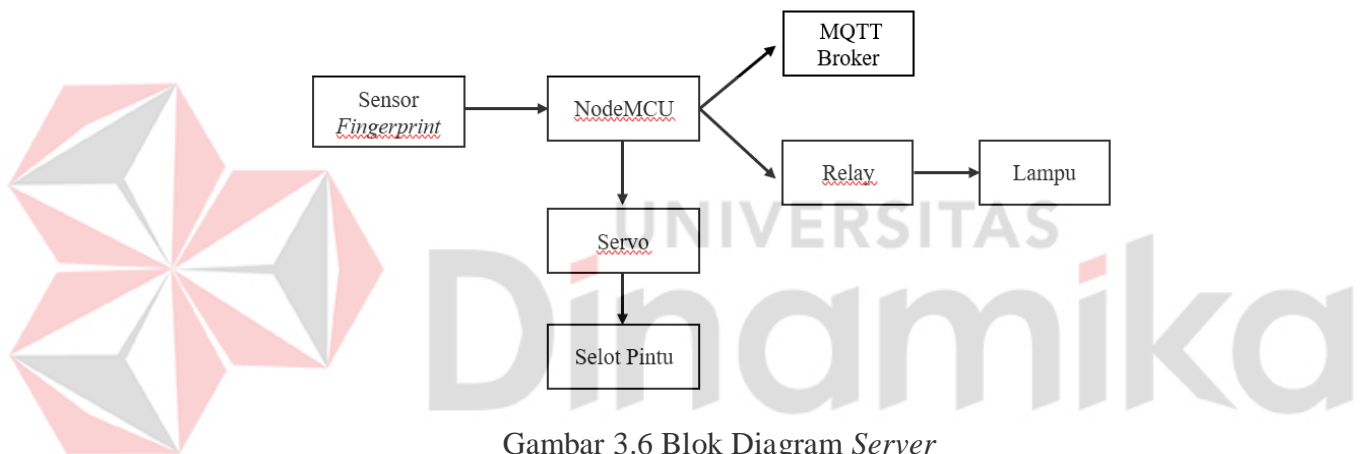
Flowchart Client 2 yang diawali dengan proses menghubungkan ke koneksi *WiFi* lokal setelah itu menghubungkan *Client 1* ke *Server* melalui *IPAddress* yang dimiliki oleh *Server* untuk proses pengiriman datanya. Input dari *Client 2* adalah jumlah partikel gas yang didapatkan dari hasil deteksi sensor MQ2. Apabila terdeteksi jumlah partikel gas dengan batas yang telah ditentukan yaitu bernilai 650, maka *Client 2* akan mengirimkan sebuah pesan ke *Server* dan juga nilai partikel gas tersebut bisa dipantau di serial monitor.

3.1.3. Server & Publisher

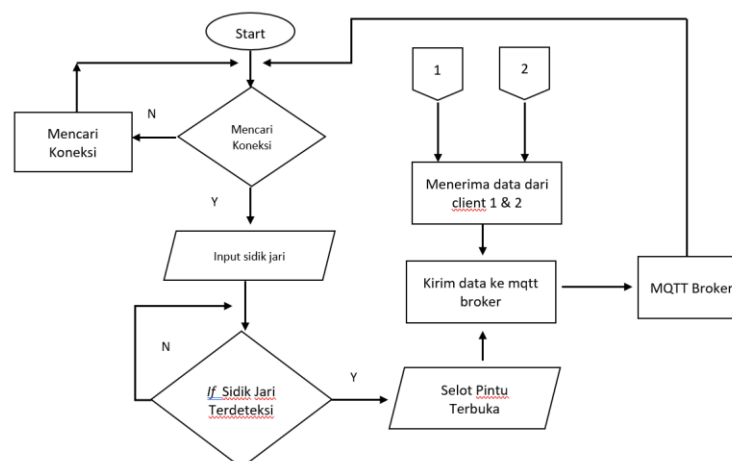
Pada Tugas Akhir ini terdapat beberapa fungsi yang dimiliki oleh *Server*, diantaranya:

1. Sebagai penerima pesan berupa data yang dikirimkan oleh kedua *Client* dan disimpan sementara sebelum dikirimkan ke perangkat *user*.
2. Sebagai (*Publisher*) atau mengirimkan pesan data yang didapat dari kedua *Client* dan kemudian dikirimkan ke MQTT Broker.
3. Sebagai penghubung ke MQTT Broker agar pesan data tersebut sampai dan diterima oleh *user*.

Flowchart dan blok diagram dari *Server* ditunjukkan Pada Gambar 3.6 dan 3.7.



Gambar 3.6 Blok Diagram *Server*

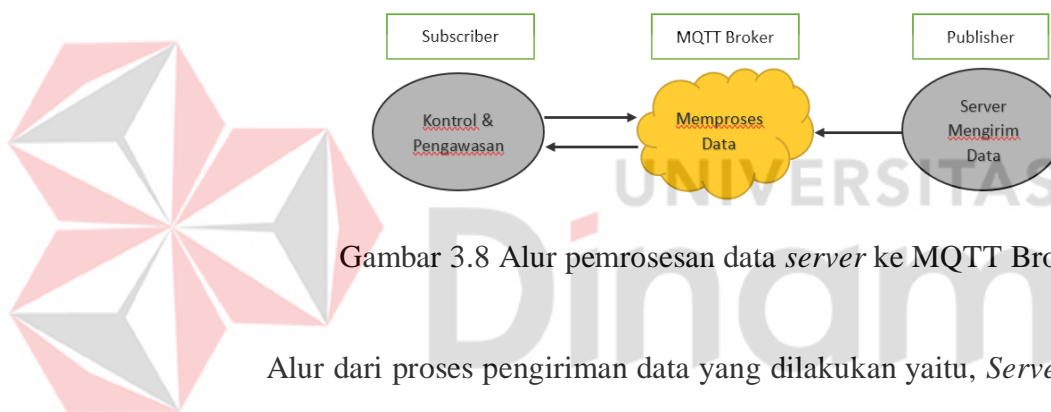


Gambar 3.7 *Flowchart Server*

Alur dari *flowchart Server* diawali dengan proses mencari koneksi *WiFi* lokal atau rumah agar bisa dihubungkan dengan *Client* dan MQTT Broker. *Server* memiliki sebuah input berupa sidik jari melalui sensor *Fingerprint*. Apabila sidik jari sudah terdeteksi, maka selot pintu akan terbuka yang digerakkan menggunakan servo. Pada koneksi *WiFi* jika sudah terhubung, maka bisa dipastikan data dari *Client* diterima dan langsung dikirimkan ke MQTT Broker.

3.1.4. MQTT Broker

Pada Tugas Akhir ini MQTT Broker bertugas seperti halnya *Server* yang bertugas menerima dan mengirimkan data, akan tetapi *Server* NodeMCU tidak bisa mengirimkan secara langsung ke *Subscriber (user)*. Alur dari pemrosesan data dari *Server* ke MQTT Broker ditunjukkan pada Gambar 3.8.



Gambar 3.8 Alur pemrosesan data *server* ke MQTT Broker

Alur dari proses pengiriman data yang dilakukan yaitu, *Server* mengirimkan data – data yang didapat dari *Client* ke Broker kemudian data tersebut akan diterima dan diproses oleh Broker. Sebelum proses pengiriman tersebut terjadi, ada suatu syarat agar pengiriman tersebut sampai di Broker yaitu dengan mengatur *topic*, *username*, *password*, *port* dan *mqttserver* pada *Server* harus sama dengan yang dimiliki oleh Broker. Apabila syarat tersebut sudah terpenuhi, maka data dari *Server* akan berhasil diterima oleh Broker. Selanjutnya Broker akan mengirimkan data tersebut ke *Subscriber*. Ketika sudah diterima oleh *Subscriber*, *Subscriber* juga bisa mem-*publish* sebuah data dengan menggunakan sistem aplikasi pada *Subscriber*. Broker juga menerima dan memproses data tersebut lalu dikirimkan ke *Server* berupa output yang diinginkan.

3.1.5. Subscriber 1

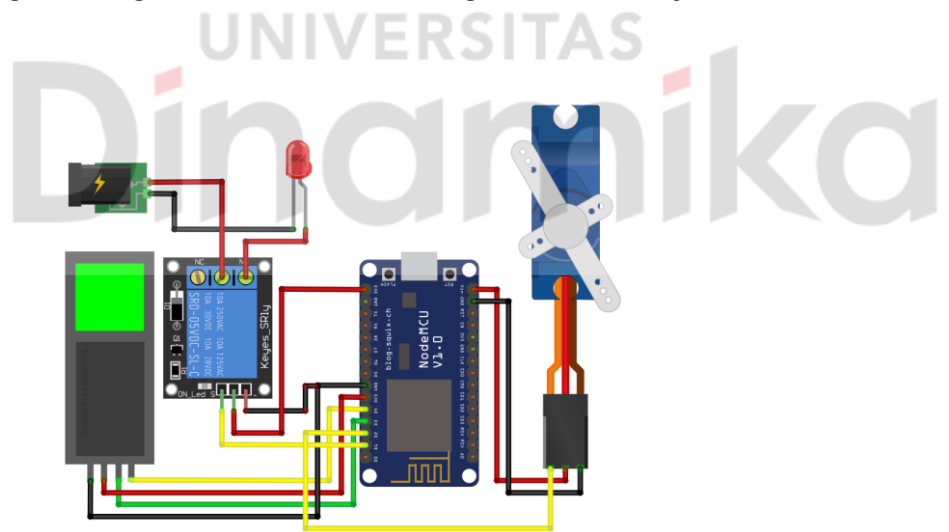
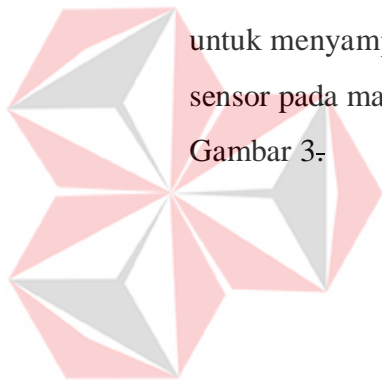
Bertugas sebagai mengawasi atau monitoring dari mqtt broker apakah pesan dari *Publisher* sudah masuk atau tidak.

3.1.6. Subscriber 2

Bertugas sebagai mengawasi atau monitoring dari MQTT Broker apakah pesan dari *Publisher* sudah masuk atau tidak dan mengontrol menyalakan lampu melalui *smartphone*.

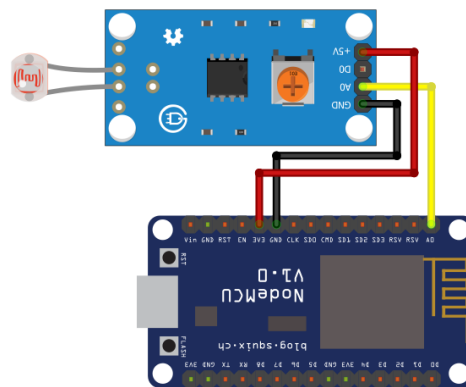
3.2 Perancangan Perangkat Keras

Pada Tugas Akhir ini memakai 3 NodeMCU dengan 2 diantaranya sebagai *Client* dan 1 lagi sebagai *Server*. *Server* disini berfungsi untuk menerima pesan dari kedua *Client* dan kemudian mengirimkannya ke *MQTT Broker*. *Client* berfungsi untuk menyampaikan sebuah pesan yang sudah diproses dengan mendapatkan dari sensor pada masing – masing *Client*. Skematik rancangan *Server* ditunjukkan Pada Gambar 3-

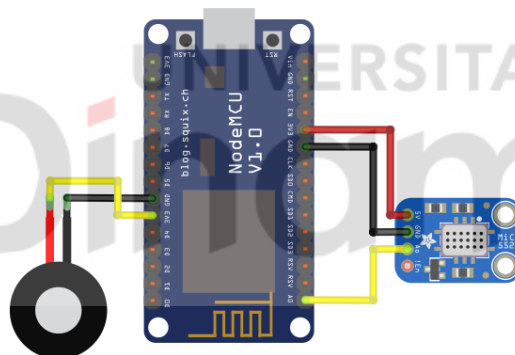


Gambar 3.9 Skematik Server

Perangkat keras pada *Server* yakni terdiri dari *Fingerprint*, Relay, LED, NodeMCU dan Servo. Sensor *Fingerprint* digunakan untuk menggerakkan Servo agar slot pintu terbuka. Lalu Relay disini berperan sebagai output dari input user yang dijalankan pada *Smartphone*.

Gambar 3.10 Skematik *Client 1*

Gambar 3.10 adalah skematik *Client 1* yang komponennya terdiri dari sensor LDR dan NodeMCU. Sensor LDR digunakan untuk mendeteksi intensitas cahaya dan apabila sudah terdeteksi sesuai batas nilai yang diberikan yaitu 512, maka *Client 1* akan mengirimkan pesan pada *Server*.

Gambar 3.11 Skematik *Client 2*

Perangkat keras pada Gambar 3.11 yaitu skematik *Client 2* yang komponennya terdiri dari sensor MQ-2 dan NodeMCU. Sensor LDR digunakan untuk mendeteksi jumlah partikel gas dan apabila sudah terdeteksi sesuai batas nilai yang diberikan yaitu 650, maka *Client 2* akan mengirimkan pesan pada *Server* dan juga buzzer akan berbunyi sebagai alarm di dalam rumah.

3.3 Perancangan *Prototype*



Gambar 3.12 Desain Pengemasan *Prototype*

Pada Gambar 3.12 adalah bentuk dari pengemasan *prototype* alat Tugas Akhir ini.



Gambar 3.13 Rangkaian alat di ruang tamu



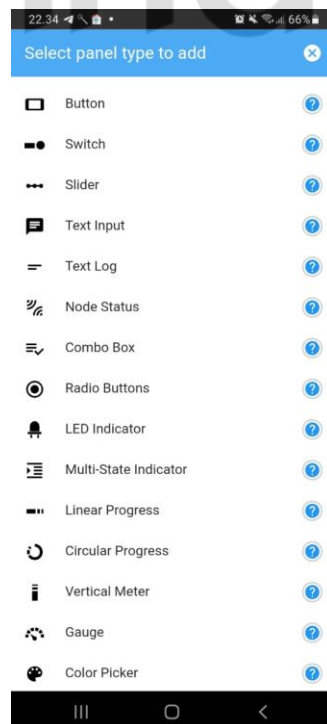
Gambar 3.14 Rangkaian alat di kamar



Gambar 3.15 Rangkaian alat di dapur

3.4.1. MQTT Panel

MQTT Panel adalah sebuah aplikasi *Smartphone* berfungsi sebagai monitoring dan kontrol sistem *Smart Home*. Ada banyak fitur panel yang dapat dibuat apabila ingin membuat sebuah sistem sendiri seperti yang ditunjukkan pada Gambar 3.16.



Gambar 3.16 Daftar fitur MQTT Panel

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Pengiriman dan Penerimaan Pesan ke *MQTT Broker* Dari *Server*

4.1.1. Tujuan

Tujuan pengujian untuk menentukan penerimaan pesan oleh *MQTT Broker* dari *Server* telah berhasil.

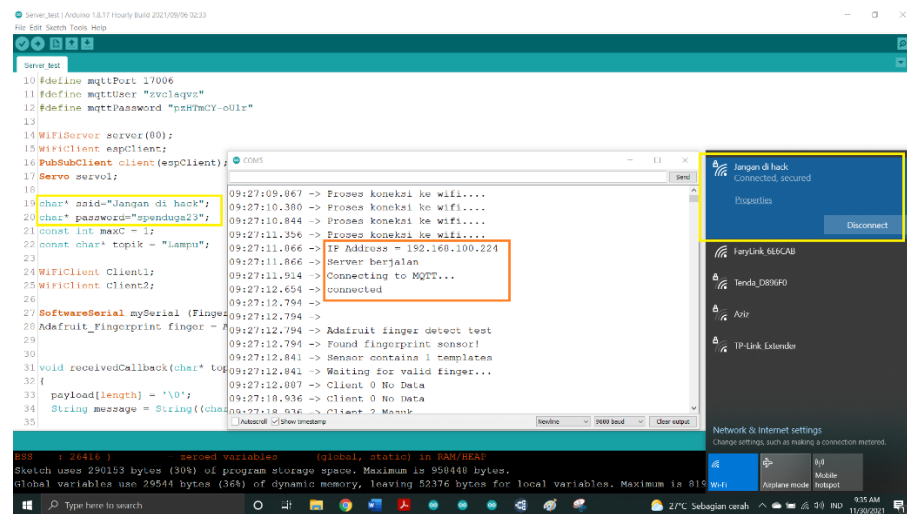
4.1.2. Perlengkapan Yang Dipakai

1. NodeMCU
2. Laptop untuk menjalankan Arduino IDE dan *MQTT Explorer*
3. Koneksi *WiFi*

4.1.3. Cara Pengujian

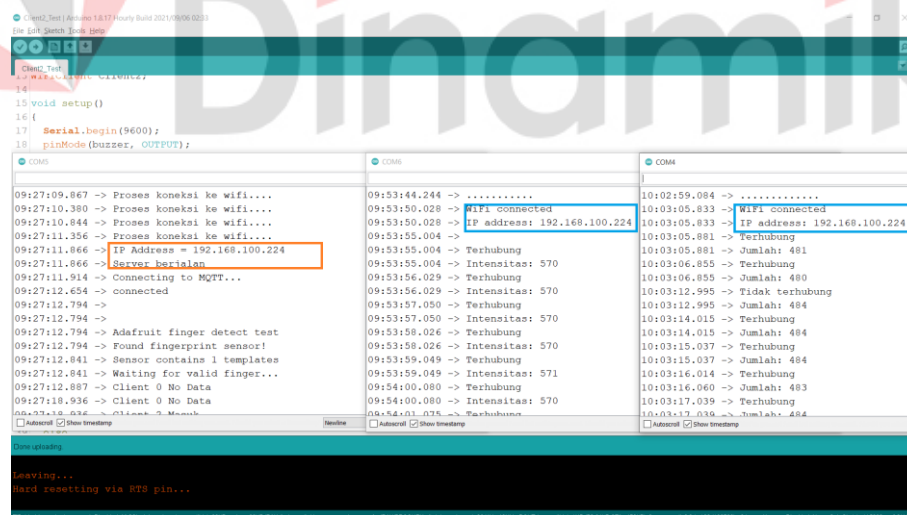
1. Menghubungkan NodeMCU dengan Laptop untuk mengupload kode dari Arduino IDE ke NodeMCU.
2. Memastikan NodeMCU (*Server*) terhubung ke *WiFi* yang sama dengan *MQTT Broker*.
3. Menghubungkan komunikasi NodeMCU (*Client 1 & Client 2*) ke *Server* melalui *IP Address Server* agar pesan dapat tersampaikan.
4. Mengamati hasil komunikasi melalui aplikasi *MQTT Explorer* dan Serial Monitor *Server*.

4.1.4. Hasil Uji



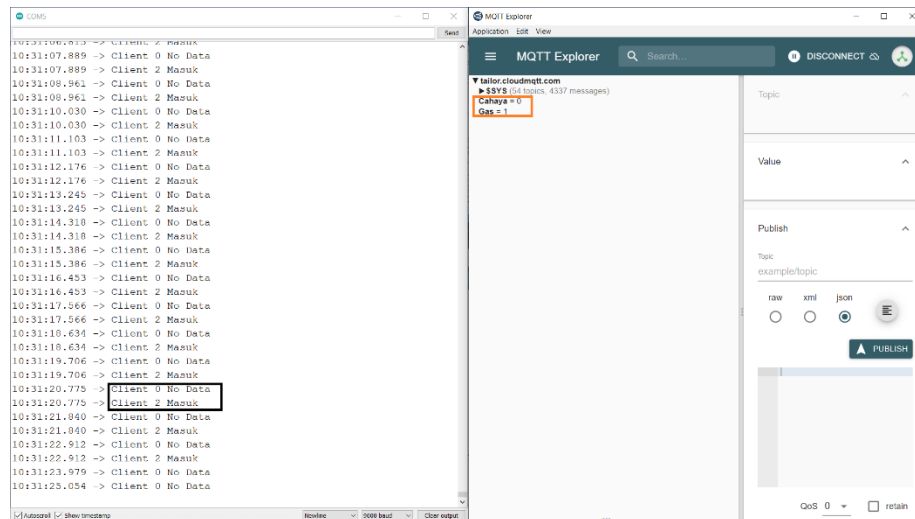
Gambar 4.1 Koneksi Server ke WiFi

Pada Gambar 4.1 dapat dilihat bahwa SSID *Server* dan SSID *WiFi* terhubung dan berhasil terkoneksi dan juga sudah terkoneksi dengan MQTT Broker.



Gambar 4.2 Koneksi Client 1 & Client 2 ke Server

Pada Gambar 4.2 adalah proses koneksi dari *Client 1* dan *Client 2* ke *Server* melalui *IP Address Server* (192.168.100.224) yang mana proses ini sangat penting agar pesan bisa diterima oleh *Server*.



Gambar 4.3 Hasil Pengamatan Pada Serial Monitor *Server* dan *MQTT Explorer*

Pada Gambar 4.3 menunjukkan hasil pengamatan pada serial monitor *Server* dan *MQTT Explorer*. Pada serial monitor dengan tanda *Client 0* adalah Cahaya di *MQTT Explorer* dan *Client 2* adalah Gas di *MQTT Explorer*.

1. Hasil Pengamatan pada Serial Monitor *Client 1*, *Client 2* dan *Server*

Tabel 4.1 Hasil pengujian *Client 1* dan *Client 2* dengan *Server*

No.	Waktu			Serial Monitor			Hasil
	Client 1	Client 2	Server	Client 1 Intensitas	Client 2 Jumlah	Server	
1	11:54:14.760	11:54:14.374	11:54:14.482	580	397	Masuk	Tidak Berhasil
2	11:54:15.781	11:54:15.348	11:54:15.827	580	397	Masuk	Berhasil
3	11:54:16.805	11:54:16.370	11:54:16.899	580	397	No Data	Berhasil
4			11:54:16.899	580	397	Masuk	Berhasil
5	11:54:17.781	11:54:17.396	11:54:18.291	580	397	No Data	Berhasil
6			11:54:18.291	580	397	Masuk	Berhasil
7	11:54:18.806	11:54:18.370	11:54:19.361	581	397	No Data	Berhasil
8			11:54:19.361	580	397	Masuk	Berhasil
9	11:54:19.826	11:54:19.391	11:54:20.710	580	396	No Data	Berhasil
10			11:54:20.710	580	397	Masuk	Berhasil
11	11:54:20.802	11:54:23.026	11:54:21.783	580	397	No Data	Berhasil
12			11:54:21.783	580	397	Masuk	Berhasil
13	11:54:26.826	11:54:29.057	11:54:23.089	580	397	No Data	Tidak Berhasil
14	11:54:30.505	11:54:30.088	11:54:24.181	580	397	No Data	Tidak Berhasil
15	11:54:31.527	11:54:31.061	11:54:30.366	581	396	Masuk	Tidak Berhasil
16	11:54:30.505	11:54:30.088	11:54:31.481	581	397	Masuk	Berhasil

17			11:54:32.828	581	397	No Data	Berhasil
18	11:54:32.501	11:54:32.086	11:54:32.828	581	396	Masuk	Berhasil
19			11:54:33.892	581	397	No Data	Berhasil
20	11:54:33.523	11:54:33.103	11:54:33.892	581	397	Masuk	Berhasil
21			11:54:35.241	581	397	No Data	Berhasil
22	11:54:34.544	11:54:34.077	11:54:35.241	581	397	Masuk	Berhasil
23			11:54:36.355	581	396	No Data	Berhasil
24	11:54:35.565	11:54:37.748	11:54:36.355	581	397	Masuk	Berhasil
25	11:54:34.544	11:54:34.077	11:54:37.657	581	397	No Data	Tidak Berhasil
26	11:54:36.540	11:54:38.770	11:54:38.770	581	397	No Data	Tidak Berhasil
27	11:54:42.584	11:54:44.773	11:54:46.032	582	397	Masuk	Tidak Berhasil
28	11:54:47.194	11:54:46.824	11:54:47.378	582	396	Masuk	Berhasil
29			11:54:48.450	582	397	No Data	Berhasil
30	11:54:48.216	11:54:47.798	11:54:48.450	582	396	Masuk	Berhasil

Pada Tabel 4.1 merupakan hasil pengujian penerimaan pesan oleh *Server* dari *Client 1* dan *Client 2*. Pesan yang dikirimkan oleh kedua *Client* yaitu berasal dari pemindaian data dari sensor LDR dan MQ2 yang terhubung dengan NodeMCU kemudian hasil pemindaian tersebut dikirimkan ke *Server*. Hasil pengujian menggunakan 30 sampel data 23 data diantaranya berhasil dan 7 lainnya tidak berhasil dikarenakan terdapat 2 data masuk sekaligus, apabila di persentasekan menjadi 76.67% untuk tingkat keberhasilan dengan indeks berhasil dan 23.33% tidak berhasil.

2. Hasil Pengiriman pesan dari *Server* ke MQTT Broker

Tabel 4.2 Hasil Pengujian Pengiriman Pesan dari *Server* ke MQTT Broker

No.	Waktu		Pesan	
	Server	MQTT Broker	Server	MQTT Broker
1	14:12:01.477	14:12:02	No Data	0
2	14:12:02.546	14:12:03	No Data	0
3	14:12:03.615	14:12:04	No Data	0
4	14:12:04.729	14:12:05	No Data	0
5	14:12:05.804	14:12:06	No Data	0
6	14:12:06.868	14:12:07	No Data	0
7	14:12:07.898	14:12:08	No Data	0
8	14:12:09.014	14:12:09	No Data	0
9	14:12:10.077	14:12:10	No Data	0
10	14:12:11.150	14:12:11	No Data	0
11	14:12:12.221	14:12:12	No Data	0
12	14:12:13.291	14:12:13	No Data	0
13	14:12:14.359	14:12:14	No Data	0
14	14:12:15.431	14:12:15	No Data	0
15	14:12:16.547	14:12:16	No Data	0
16	14:12:17.578	14:12:17	No Data	0
17	14:12:18.693	14:12:18	No Data	0

18	14:12:19.758	14:12:19	No Data	0
19	14:12:20.831	14:12:20	No Data	0
20	14:12:21.898	14:12:21	No Data	0
21	14:12:22.961	14:12:22	No Data	0
22	14:12:24.030	14:12:23	No Data	0
23	14:12:25.098	14:12:24	No Data	0
24	14:12:26.169	14:12:25	No Data	0
25	14:12:27.285	14:12:26	No Data	0
26	14:12:28.352	14:12:27	No Data	0
27	14:12:29.421	14:12:28	No Data	0
28	14:12:30.490	14:12:29	No Data	0
29	14:12:31.558	14:12:30	No Data	0
30	14:12:32.587	14:12:31	No Data	0

Tabel 4.2 menyajikan hasil pengujian penerimaan pesan yang dikirim oleh *Server* ke MQTT broker. Diambil salah satu contoh *topic* Cahaya dengan pesan yang dikirim dari *Server* yaitu 0 dan broker menerima pesan sama dengan 0. No Data pada tabel adalah hanya tampilan semata pada serial monitor yang didapat pesannya dari *Client 1*, akan tetapi pesan yang dikirim dari *Server* tetap 0. Hasil pengujian dengan 30 sampel data berhasil diterima oleh MQTT Broker dengan tingkat keberhasilan 100%. Ada beberapa pengiriman pesan dari *Server* terdapat *delay* dikarenakan koneksi *Client* ada yang terputus, tapi tidak lama kemudian *Client* akan terhubung kembali dan mengirimkan data ke *Server* lagi. Tampilan kode pesan yang dikirim dari *Server* ditunjukkan pada Gambar 4.4 ditandai dengan kotak hitam.



```

137
138 Client1 = server.available();
139 String pesan1 = Client1.readStringUntil('\n');
140 if (pesan1.charAt(0) == '1')
141 {
142     Serial.println("Client " + pesan1);
143     client.publish("Cahaya", "1");
144 }
145 else if (pesan1.charAt(0) == '0')
146 {
147     Serial.println("Client " + pesan1);
148     client.publish("Cahaya", "0");
149 }
150
151 Client2 = server.available();
152 String pesan2 = Client2.readStringUntil('\n');
153 if (pesan2.charAt(0) == '2')
154 {
155     Serial.println("Client " + pesan2);
156     client.publish("Gas", "1");
157 }
158 else if (pesan2.charAt(0) == '3')
159 {
160     Serial.println("Client " + pesan2);
161     client.publish("Gas", "0");
162 }

```

Gambar 4.4 Kode pesan dari *Server*

4.2 Pengujian Aplikasi mobile

4.2.1 Tujuan

Tujuan pengujian pada aplikasi mobile ini diperuntukkan *monitoring* dan kontrol. Pengawasan atau *monitoring* pada aplikasi MQTT Panel yang ditandai dengan 2 buah LED sebagai media output pada aplikasi tersebut. Terdapat 2 LED yaitu, 1 LED untuk Sensor LDR dan 1 LED untuk Sensor MQ2. Kedua LED akan menyampaikan informasi bahwa sensor – sensor telah mendeteksi apa yang harus di deteksi sesuai dengan program, apabila berhasil mendeteksi maka LED akan menyala. Sedangkan pengujian kontrol disini menggunakan panel *switch* yang berfungsi sebagai media input untuk mengaktifkan relay yang terhubung dengan lampu.

4.2.2 Peralatan Yang Dipakai

1. NodeMCU
2. Laptop untuk menjalankan Arduino IDE dan MQTT Explorer
3. Koneksi WiFi
4. Smartphone

4.2.3 Cara Pengujian

1. Menghubungkan *Server* dengan MQTT Broker
2. Mengamati LED 1 (Sensor LDR) dan LED 2 (Sensor MQ2)
3. Menyalakan Lampu dengan *Switch* pada MQTT Panel

4.2.4 Pengujian Monitoring

```

16:31:35.179 -> Proses koneksi ke wifi....
16:31:35.695 -> Proses koneksi ke wifi....
16:31:36.164 -> Proses koneksi ke wifi....
16:31:36.695 -> Proses koneksi ke wifi....
16:31:37.163 -> Proses koneksi ke wifi....
16:31:37.678 -> IP Address = 192.168.100.224
16:31:37.678 -> Server berjalan
16:31:37.725 -> Connecting to MQTT...
16:31:38.397 -> connected
16:31:38.491 ->
16:31:38.491 ->
16:31:38.491 -> Adafruit finger detect test
16:31:38.491 -> Found fingerprint sensor!
16:31:38.538 -> Sensor contains 1 templates
16:31:38.584 -> Waiting for valid finger...
16:31:49.661 -> Client 2 Masuk
16:31:50.740 -> Client 2 Masuk
16:31:52.114 -> Client 0 No Data
16:31:52.114 -> Client 2 Masuk
16:31:53.192 -> Client 0 No Data
16:31:53.192 -> Client 2 Masuk
16:31:54.519 -> Client 0 No Data
16:31:54.519 -> Client 2 Masuk
16:31:55.598 -> Client 0 No Data
16:31:55.598 -> Client 2 Masuk
16:31:56.956 -> Client 0 No Data
16:31:56.956 -> Client 2 Masuk
16:31:58.026 -> Client 0 No Data
16:31:58.026 -> Client 2 Masuk
16:31:59.372 -> Client 0 No Data
16:32:05.676 -> Client 0 No Data
16:32:05.676 -> Client 2 Masuk
16:32:06.737 -> Client 0 No Data
16:32:06.737 -> Client 2 Masuk
  
```

Gambar 4.5 Koneksi Server ke MQTT Broker

Pada Gambar 4.5 adalah proses menghubungkan NodeMCU Server ke MQTT Broker agar pesannya diterima dan sudah terkoneksi.

```

1 #include <PubSubClient.h>
2 #include <ESP8266WiFi.h>
3 #include <Adafruit_Fingerprint.h>
4 #include <SoftwareSerial.h>
5 #include <Servo.h>
6
7 #define Finger_Rx 0 //
8 #define Finger_Tx 2 //
9 #define mqttServer "192.168.100.224"
10 #define mqttPort 17004
11 #define mqttUser "root"
12 #define mqttPassword "root"
13
14 WiFiServer server(80);
15 WiFiClient espClient;
16 PubSubClient client(espClient);
17 Servo servo;
18
19 char* ssid="Jangsan di";
20 char* password="apandi";
21 const int maxC = 1;
22 const char* topik = "MQTT";
23
24 WiFiClient Client1;
25 WiFiClient Client2;
26
27
28
  
```

```

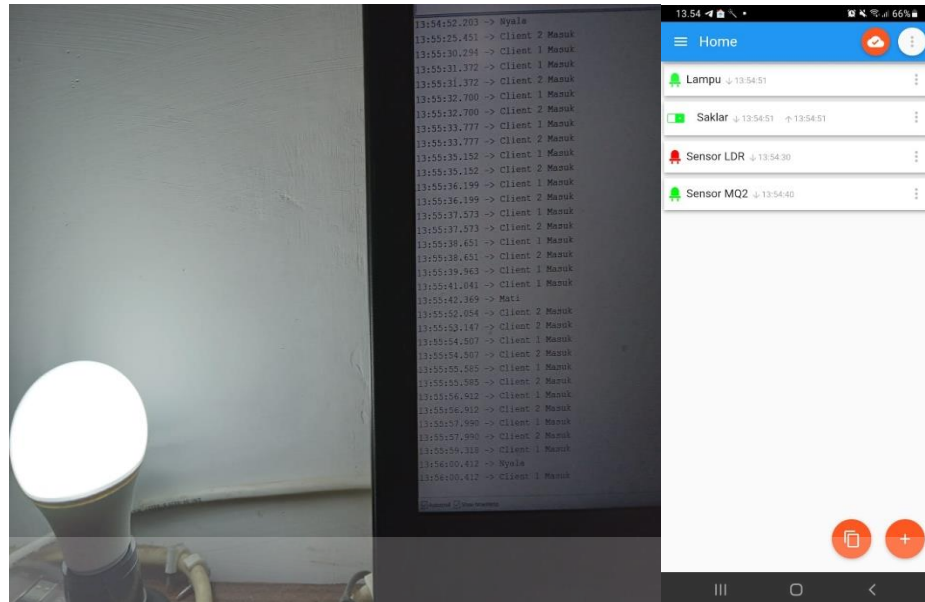
17:29:33.023 -> Client 0 No Data
17:29:51.376 -> Client 2 Masuk
17:29:59.525 -> Client 2 Masuk
17:30:01.568 -> Client 0 No Data
17:30:05.661 -> Client 2 Masuk
17:30:07.703 -> Client 0 No Data
17:30:07.703 -> Client 2 Masuk
17:30:09.756 -> Client 0 No Data
17:30:15.849 -> Client 2 Masuk
17:30:17.921 -> Client 0 No Data
17:30:17.921 -> Client 2 Masuk
17:30:19.941 -> Client 0 No Data
17:30:19.941 -> Client 2 Masuk
17:30:21.987 -> Client 0 No Data
17:30:21.987 -> Client 2 Masuk
17:30:24.034 -> Client 0 No Data
17:30:24.034 -> Client 2 Masuk
17:30:26.086 -> Client 0 No Data
17:30:26.086 -> Client 2 Masuk
17:30:28.132 -> Client 0 No Data
17:30:28.132 -> Client 2 Masuk
17:30:30.153 -> Client 0 No Data
17:30:30.153 -> Client 2 Masuk
17:30:32.200 -> Client 0 No Data
17:30:32.200 -> Client 2 Masuk
17:30:34.240 -> Client 0 No Data
17:30:34.240 -> Client 2 Masuk
17:30:36.264 -> Client 0 No Data
17:30:38.301 -> Client 0 No Data
17:30:38.301 -> Client 2 Masuk
17:30:40.345 -> Client 0 No Data
17:30:40.345 -> Client 2 Masuk
  
```

Gambar 4.6 Hasil Monitoring pada Aplikasi MQTT Panel

Gambar 4.6 merupakan hasil pengamatan pesan yang dikirim dari Server 2 buah pesan masuk *Client 0* No Data pada serial monitor dan LED sensor LDR berwarna merah adalah hasil dari pesan tersebut. Sedangkan *Client 2* Masuk pada serial monitor dan LED sensor MQ2 berwarna hijau adalah hasil dari pesan yang

telah masuk dari *Server*.

4.2.5 Pengujian Controlling



Gambar 4.7 Hasil kontrol lampu dengan aplikasi MQTT panel

Bisa dilihat Pada Gambar 4.7 hasil dari kontrol dengan cara menekan tombol *switch* pada aplikasi MQTT Panel seperti di Gambar 4.7 dan hasilnya lampu menyala dengan sempurna.

4.3 Pengujian Transmisi Data dengan Protokol MQTT

4.3.1 Tujuan

Tujuan dari pengujian ini yaitu untuk mengetahui waktu *delay* dari pesan yang dikirimkan oleh *Server* ke MQTT Broker.

4.3.2 Peralatan Yang Dipakai

1. NodeMCU
2. Laptop untuk menjalankan Arduino IDE dan MQTT Explorer
3. Koneksi *WiFi*

4.3.3 Cara Pengujian

1. Menghubungkan *Server* dengan MQTT Broker

2. Mengamati dan mengambil data waktu saat *Server* mengirim pesan ke MQTT Broker.
3. Melakukan perhitungan *delay* dari data sebelumnya.

4.3.4 Hasil Pengujian

Tabel 4.3 Hasil Pengujian *delay Server* dengan MQTT Broker

Delay (s)			
Waktu			
No.	Server (s)	MQTT Broker (s)	Hasil(s)
1	14:12:01.477	14:12:02	0.523
2	14:12:02.546	14:12:03	0.454
3	14:12:03.615	14:12:04	0.385
4	14:12:04.729	14:12:05	0.271
5	14:12:05.804	14:12:06	0.196
6	14:12:06.868	14:12:07	0.132
7	14:12:07.898	14:12:08	1.102
8	14:12:09.014	14:12:09	0.986
9	14:12:10.077	14:12:10	0.923
10	14:12:11.150	14:12:11	0.850
11	14:12:12.221	14:12:12	0.779
12	14:12:13.291	14:12:13	0.709
13	14:12:14.359	14:12:14	0.641
14	14:12:15.431	14:12:15	0.569
15	14:12:16.547	14:12:16	0.453
16	14:12:17.578	14:12:17	0.422
17	14:12:18.693	14:12:18	0.307
18	14:12:19.758	14:12:19	0.242
19	14:12:20.831	14:12:20	0.169
20	14:12:21.898	14:12:21	0.102
21	14:12:22.961	14:12:22	1.039
22	14:12:24.030	14:12:23	0.97
23	14:12:25.098	14:12:24	0.902
24	14:12:26.169	14:12:25	0.831
25	14:12:27.285	14:12:26	0.715
26	14:12:28.352	14:12:27	0.648
27	14:12:29.421	14:12:28	0.579
28	14:12:30.490	14:12:29	0.51
29	14:12:31.558	14:12:30	0.442

30	14:12:32.587	14:12:31	0.413
31	14:12:33.700	14:12:32	-
32		14:12:33	-
Nilai Rata – rata			0.098

Pada Tabel 4.3 waktu pertama yang ditunjukkan oleh serial monitor adalah 14:12:01.477 sedangkan waktu pertama yang ditunjukkan oleh mqtt broker adalah 14:12:01.000 terdapat perbedaan waktu dikedua tempat sebesar 0.477s. Sehingga rata – rata delay menjadi : $0.575s - 0.477s = 0.098s$. Hal ini termasuk dalam kategori delay sangat bagus sesuai tabel 3.1, dikarenakan memiliki nilai $< 0.15s$.

4.4 Hasil Pembahasan

1. Pengujian pengiriman dan penerimaan data ke MQTT Broker dari *Server* berhasil 100% dengan melalui pengambilan 30 sampel data. Data tersebut tercantum pada Tabel 4.1 dan Tabel 4.2 yang diawali dengan proses menghubungkan koneksi *Client 1* dan *Client 2* dengan *Server* yang memiliki IP 192.168.100.224, pesan dari kedua *Client* berhasil diterima oleh *Server*. Kemudian pada MQTT Broker sudah dapat menerima pesan yang dikirimkan oleh *Server* dan bisa dipantau melalui aplikasi MQTT Explorer.
2. Pengujian monitoring dan kontrol melalui aplikasi *smartphone* MQTT Panel sudah berhasil melalui proses pengiriman pesan dari *Server* ke MQTT Broker. Indikasi keberhasilan monitoring pada MQTT Panel ditandai dengan waktu yang berganti terus menerus dan LED ungu dan hijau berarti “ON” lalu LED merah berarti “OFF”. Sedangkan untuk indikasi keberhasilan kontroling pada MQTT Panel ditandai dengan lampu kamar menyala.
3. Pengujian *delay* pengiriman data *Server* ke broker melalui proses perhitungan dengan 30 sampel data dan didapatkan dengan nilai rata – rata 0.509ms yang masuk dalam kategori buruk dalam proses pengiriman.

BAB V

PENUTUP

5.1. KESIMPULAN

Adapun kesimpulan yang didapatkan dari hasil pada Tugas Akhir ini adalah sebagai berikut :

1. Sistem dari *smart home* yang disusun oleh 3 NodeMCU dapat terhubung dengan pusat 1 NodeMCU sebagai *Server* dan pesan dari kedua *Client* dapat diterima dengan tingkat keberhasilan 76.67%.
2. Penerapan protokol MQTT pada sistem *smart home* agar 3 NodeMCU berhasil diterapkan karena pesan dari NodeMCU sudah dapat diterima oleh MQTT Broker sebagai *Server* dengan tingkat keberhasilan 100% dengan rata-rata *delay* 0.098s yang termasuk dalam kategori sangat bagus.
3. Sistem monitoring sudah dapat menerima informasi dengan indikasi keberhasilan 100% dan kontroling dengan aplikasi MQTT Panel pada *Smartphone* telah berhasil menyalakan dan mematikan lampu 100%.

5.2. SARAN

Adapun saran untuk penelitian ini sebagai berikut :

1. Melakukan pengembangan pada bagian *Client 2* yang menggunakan sensor LDR dengan memberi timer (*Real Time Clock*) agar lampu bisa menyala secara otomatis.
2. Melakukan pembembangan fitur *multiclient*, apabila saat ingin menambahkan *client* tanpa merubah kode atau *source code* di *server*.

DAFTAR PUSTAKA

- Aditya, F. G., Hafidudin & Permana, G. A., 2015. ANALISIS DAN PERANCANGAN PROTOTYPE *SMART HOME* DENGAN SISTEM. *e-Proceeding of Engineering*, p. 3071.
- Arsyad, O. R., Kurnia & Kartika, P., 2021. RANCANG BANGUN ALAT PENGAMAN BRANKAS MENGGUNAKAN SENSOR SIDIK JARI BERBASIS ARDUINO. *Jurnal Mahasiswa Teknik Informatika*.
- Ashari, M. A. & Lidyawati, L., 2019. IOT BERBASIS SISTEM *SMART HOME* MENGGUNAKAN. *Jurnal Kajian Teknik Elektro*, III(2), pp. 67-172.
- Pr., E., 2015 [Online].
Available at: <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>
[Accessed 22 11 2021].
- Rinaldy, Christiani, R. F. & Supriyadi, D., 2013. Pengendalian Motor Servo Yang Terintegrasi Dengan Webcam Berbasis Internet Dan Arduino. *Jurnal Infotel*, V(2).
- Saputra, G. Y. et al., 2017. PENERAPAN PROTOKOL MQTT PADA TEKNOLOGI WAN. *Informatika Mulawarman*, 12(2).
- Sarmidi & Fauzi, R. A., 2019. Pendeteksi Kebocoran Gas Menggunakan Sensor MQ-2 Berbasis Arduino UNO. *Manajemen dan Teknik Informatika*, pp. 51-60.
- Tsauqi, A. K. et al., 2016. SAKLAR OTOMATIS BERBASIS LIGHT DEPENDENT RESISTOR (LDR). *E-Journal*, Volume V.
- Widodo, Y. B., Ichsan, A. M. & Sutabri, T., 2020. Perancangan Sistem *Smart Home* Dengan Konsep Internet Of Things Hybrid Berbasis Protokol Message Queuing Telemetry Transport. *Jurnal Teknologi Informatika dan Komputer MH Thamrin*, 6(2).