

Rancang Bangun Sistem Monitoring Dan Kontrol *Smart Home* Menggunakan Protokol MQTT

Eka Fijianara Fernanda¹⁾ Ira Puspasari²⁾ Harianto³⁾ Musayyanah⁴⁾

Program Studi/Jurusan Teknik Komputer
Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1) ekafuji754@gmail.com, 2) ira@dinamika.ac.id, 3) harianto@dinamika.ac.id, 4) musay@dinamika.ac.id

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 di rumah 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

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 dapat 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

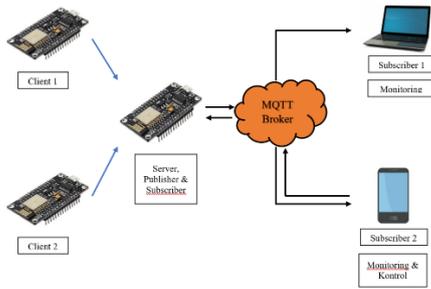
multi client dengan protokol MQTT. Kekurangan dari penelitian dari tersebut, yaitu mikrokontroler NodeMCU yang digunakan hanya satu yang menyebabkan I/O tidak dapat terlalu banyak (Widodo, et al., 2020).

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 dapat dimonitoring dan kontrol menggunakan protokol MQTT. Kelebihan yang dimiliki oleh penelitian ini yaitu menggunakan 3 buah mikrokontroler agar dapat 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.

METODE PENELITIAN

Model Perancangan

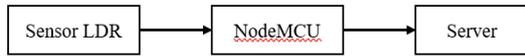
Gambar 1 menunjukkan ada beberapa bagian dari model perancangan sistem yang memiliki fungsi tersendiri, berikut adalah daftar dan penjelasan dari setiap bagiannya.



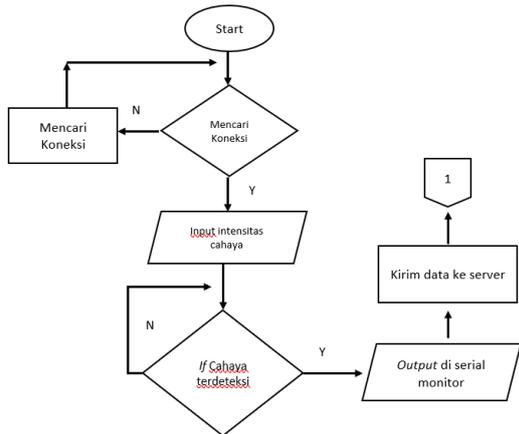
Gambar 1. Model perancangan sistem

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 dapat membedakan asal dari pesan tersebut. Flowchart dan blok diagram dari Client 1 ditunjukkan pada gambar 2 dan 3.



Gambar 2. Blok Diagram Client 1

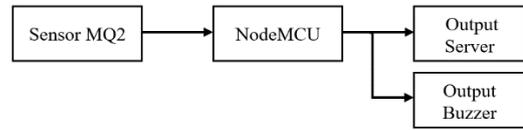


Gambar 3. Flowchart Client 1

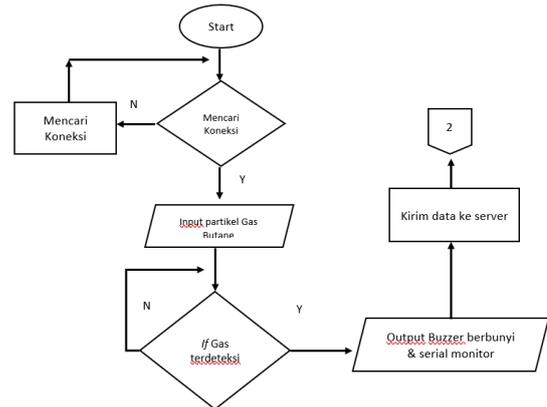
Client 2

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 dapat membedakan asal dari pesan tersebut. Flowchart dan blok diagram dari Client 1 ditunjukkan pada gambar 4 dan 5.



Gambar 4. Blok Diagram Client 2



Gambar 5. Flowchart Client 2

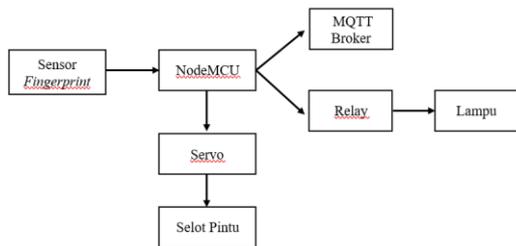
Flowchart Client 2 yang diawali dengan proses menghubungkan ke koneksi WiFi lokal setelah itu menghubungkan Client 1 ke Server melalui IP Address yang dimiliki oleh Server untuk proses pengiriman datanya. Input dari Client 2 adalah jumlah partikel gas yang didapatkan dari hasil deteksi sensor MQ2. Apabila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

Server dan Publisher

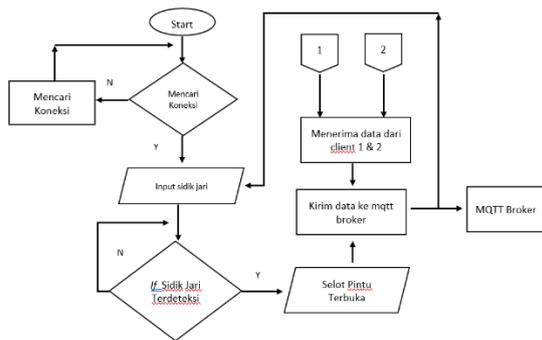
Pada penelitian ini terdapat beberapa fungsi yang dimiliki oleh Server, diantaranya:

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 6 dan 7.



Gambar 6. Blok Diagram Server

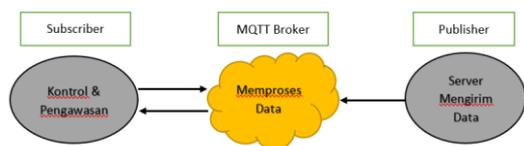


Gambar 7. Flowchart Server

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

MQTT Broker

Pada penelitian ini MQTT Broker bertugas seperti halnya Server yang bertugas menerima dan mengirimkan data, tetapi Server NodeMCU tidak dapat mengirimkan secara langsung ke Subscriber (user). Alur dari pemrosesan data dari Server ke MQTT Broker ditunjukkan pada gambar 8.



Gambar 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 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 berhasil diterima oleh Broker. Selanjutnya Broker mengirimkan data tersebut ke Subscriber. Ketika sudah diterima oleh Subscriber, Subscriber juga dapat 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.

Subscriber 1

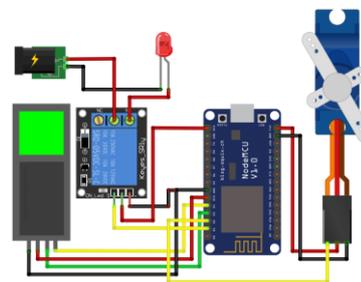
Bertugas sebagai mengawasi atau monitoring dari mqtt broker apakah pesan dari Publisher sudah masuk atau tidak.

Subscriber 2

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

Perancangan Perangkat Keras

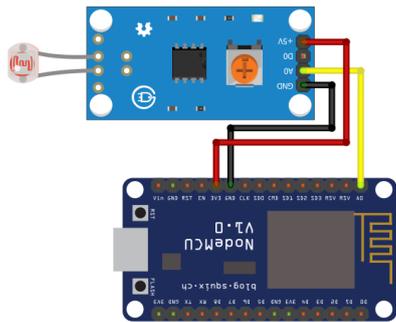
Pada penelitian ini memakai 3 NodeMCU dengan 2 diantaranya sebagai Client dan 1 lagi sebagai Server. Server disini berfungsi untuk menerima pesan dari kedua Client dan kemudian mengirimkannya ke MQTT Broker. Client berfungsi untuk menyampaikan sebuah pesan yang sudah diproses dengan mendapatkan dari sensor pada masing-masing Client. Skematik rancangan Server ditunjukkan pada gambar 9.



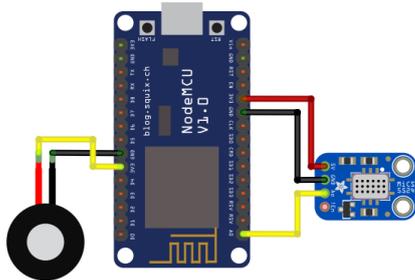
Gambar 9. Skematik Server

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

Gambar 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 mengirimkan pesan pada Server.



Gambar 10. Skematik *Client 1*



Gambar 11. Skematik *Client 2*

Perangkat keras pada gambar 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 350, maka *Client 2* mengirimkan pesan pada *Server* dan juga buzzer berbunyi sebagai alarm di dalam rumah.

Perancangan *Prototype*



Gambar 12. Desain Pengemasan *Prototype*

Pada gambar 12 adalah bentuk dari pengemasan *prototype* alat penelitian ini.



Gambar 13. Rangkaian alat di ruang tamu



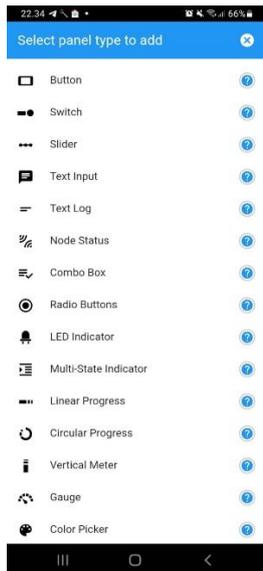
Gambar 14. Rangkaian alat di kamar



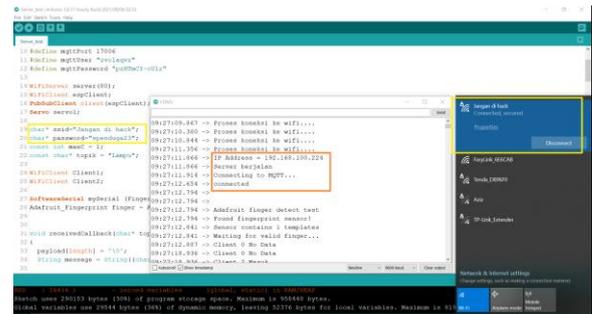
Gambar 15. Rangkaian alat di dapur

MQTT Panel

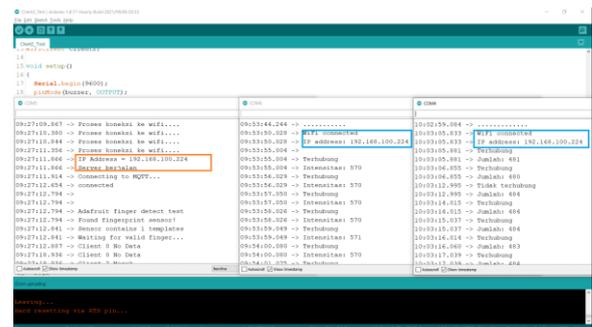
MQTT Panel adalah sebuah aplikasi *Smartphone* berfungsi sebagai monitoring dan kontrol sistem *Smart Home*. Ada banyak fitur panel yang dapat dibuat apabila ingin membuat sebuah sistem sendiri seperti yang ditunjukkan pada gambar 16.



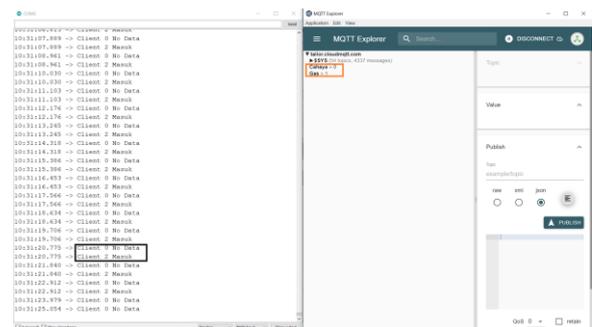
Gambar 16. Daftar fitur MQTT Panel



Gambar 17. Koneksi Server ke WiFi



Gambar 18. Koneksi Client 1 & Client 2 ke Server



Gambar 19. Hasil Pengamatan Pada Serial Monitor Server dan MQTT Explorer

HASIL DAN PEMBAHASAN

Pengujian Pengiriman dan Penerimaan Pesan ke MQTT Broker Dari Server Tujuan

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

Perlengkapan Yang Dipakai

- NodeMCU
- Laptop untuk menjalankan Arduino IDE dan MQTT Explorer
- Koneksi WiFi

Cara Pengujian

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

Hasil Uji

Pada gambar 17 dapat dilihat bahwa SSID Server dan SSID WiFi terhubung dan berhasil terkoneksi dan juga sudah terkoneksi dengan MQTT Broker.

Pada gambar 18 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 dapat diterima oleh Server.

Pada gambar 19 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.

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

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

No.	Waktu			Serial Monitor		Server	Hasil
	Client 1	Client 2	Server	Client 1 Intensitas	Client 2 Jumlah		
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			11:54:18.291	580	397	No Data	Berhasil
6	11:54:17.781	11:54:17.396	11:54:18.291	580	397	Masuk	Berhasil
7			11:54:19.361	581	397	No Data	Berhasil
8	11:54:18.806	11:54:18.370	11:54:19.361	580	397	Masuk	Berhasil
9			11:54:20.710	580	396	No Data	Berhasil
10	11:54:19.826	11:54:19.391	11:54:20.710	580	397	Masuk	Berhasil
11			11:54:21.783	580	397	No Data	Berhasil
12	11:54:20.802	11:54:23.026	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	11:54:32.501	11:54:32.086	11:54:32.828	581	397	No Data	Berhasil
18			11:54:32.828	581	396	Masuk	Berhasil
19	11:54:33.523	11:54:33.103	11:54:33.892	581	397	No Data	Berhasil
20			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	11:54:48.216	11:54:47.798	11:54:48.450	582	397	No Data	Berhasil
30			11:54:48.450	582	396	Masuk	Berhasil

Tabel 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.

Hasil Pengiriman pesan dari *Server* ke MQTT Broker

Tabel 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 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*, 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* terhubung kembali dan mengirimkan data ke *Server* lagi. Tampilan kode pesan yang dikirim dari *Server* ditunjukkan pada Gambar 20 ditandai dengan kotak hitam.



Gambar 20. Kode pesan dari *Server*

Pengujian Aplikasi mobile

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 menyampaikan informasi bahwa sensor-sensor telah mendeteksi apa yang harus di deteksi sesuai dengan program, apabila berhasil mendeteksi, maka LED menyala, sedangkan pengujian kontrol disini menggunakan panel *switch* yang berfungsi sebagai media input untuk mengaktifkan relay yang terhubung dengan lampu.

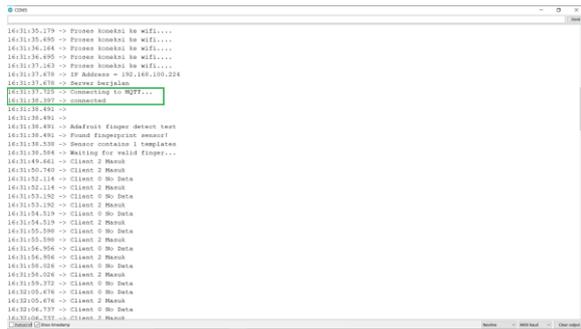
Peralatan Yang Dipakai

- NodeMCU
- Laptop untuk menjalankan Arduino IDE dan MQTT Explorer
- Koneksi WiFi
- Smartphone

Cara Pengujian

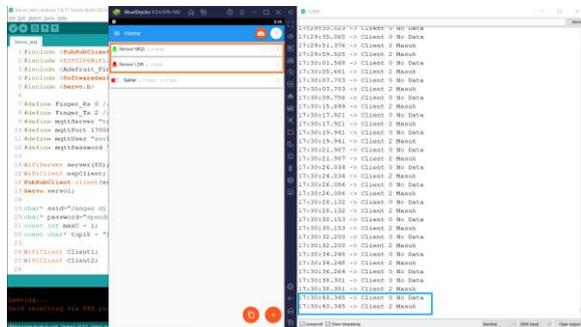
- Menghubungkan *Server* dengan MQTT Broker
- Mengamati LED 1 (Sensor LDR) dan LED 2 (Sensor MQ2)
- Menyalakan Lampu dengan *Switch* pada MQTT Panel

Pengujian Monitoring



Gambar 21. Koneksi Server ke MQTT Broker

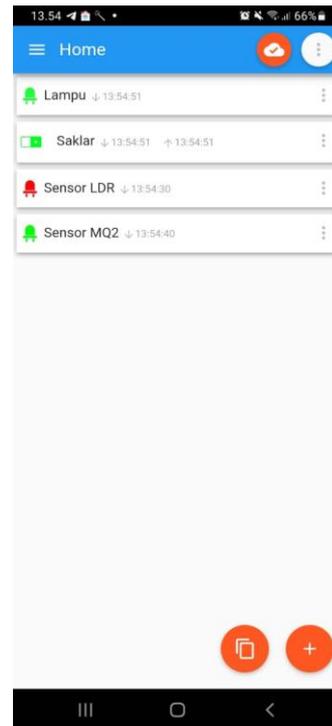
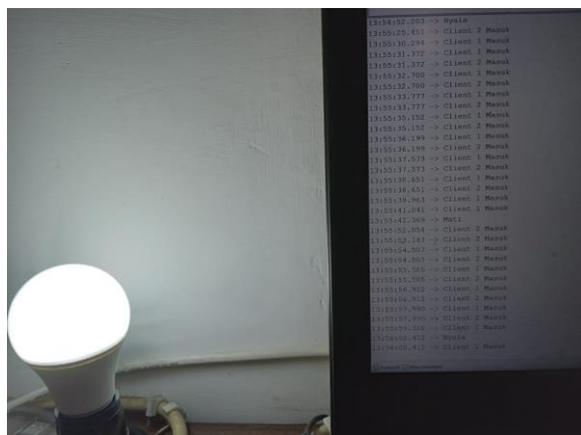
Pada gambar 21 adalah proses menghubungkan NodeMCU Server ke MQTT Broker agar pesannya diterima dan sudah terkoneksi.



Gambar 22. Hasil Monitoring pada Aplikasi MQTT Panel

Gambar 22 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.

Pengujian Controlling



Gambar 23. Hasil kontrol lampu dengan aplikasi MQTT panel

Dapat dilihat pada gambar 23 hasil dari kontrol dengan cara menekan tombol *switch* pada aplikasi MQTT Panel seperti di gambar 23 dan hasilnya lampu menyala dengan sempurna.

Pengujian Transmisi Data dengan Protokol MQTT

Tujuan

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

Peralatan Yang Dipakai

- NodeMCU
- Laptop untuk menjalankan Arduino IDE dan MQTT Explorer
- Koneksi WiFi

Cara Pengujian

- Menghubungkan Server dengan MQTT Broker
- Mengamati dan mengambil data waktu saat Server mengirim pesan ke MQTT Broker.
- Melakukan perhitungan *delay* dari data sebelumnya.

Hasil Pengujian Transmisi Data dengan Protokol MQTT

Tabel 3. Hasil Pengujian *delay Server* dengan MQTT Broker

No.	Server (s)	Delay (s) Waktu	
		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 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, dikarenakan memiliki nilai $< 0.15s$.

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 1 dan tabel 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 dapat 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.

KESIMPULAN

Adapun kesimpulan yang didapatkan dari hasil pada penelitian 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%.

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 dapat dapat 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).