PERANCANGAN SISTEM PEMANTAUAN KEBISINGAN BERBASIS IOT DENGAN METODE KOMUNIKASI HALF-DUPLEX MENGGUNAKAN ESP32 DI MASJID FATHUN QARIB DAN PERPUSTAKAAN UIN AR-RANIRY

TUGAS AKHIR

Diajukan oleh:

IRFANI ARIJA

NIM. **210705110**

Mahasiswa Fakultas Sains Dan Teknologi

Program Studi Teknologi Informasi

-

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
1445 H/ 2024 M

LEMBAR PERSETUJUAN

PERANCANGAN SISTEM PEMANTAUAN KEBISINGAN BERBASIS IOT DENGAN METODE KOMUNIKASI HALF-DUPLEX MENGGUNAKAN ESP32 DI MASJID FATHUN QARIB DAN PERPUSTAKAAN UIN AR-RANIRY

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu/Prodi Teknologi Informasi

Oleh:

IRFANI ARIJA

210705110

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Mulkan Fadhli, M.T	**Muhammad Syamsu Rizal, M.T**
NIP. 198811282020121006	

Mengetahui,
Ketua Program Studi Teknologi Informasi

Malahayati, M.T
NIP. 198301272015032003

LEMBAR PENGESAHAN

PERANCANGAN SISTEM PEMANTAUAN KEBISINGAN BERBASIS IOT DENGAN METODE KOMUNIKASI HALF-DUPLEX MENGGUNAKAN ESP32 DI MASJID FATHUN QARIB DAN PERPUSTAKAAN UIN AR-RANIRY

TUGAS AKHIR

Telah Diuji Oleh Dewan Penguji Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu Teknologi Informasi

Pada Hari/Tanggal: Jumat, 3 Januari 2025 M

3 Rajab 1446 H

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,	Sekretaris,
Mulkan Fadhli,M.T	**Muhammad Syamsu Rizal,M.T**
NIP. 198811282020121006	
Penguji I,	Penguji II,
Malahayati, M. T	**Baihaqi,M.T**
NIP.198301272015032003	**NIP. 198802212022031001**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir Muhammad Dirhamsyah, M.T., I. P. U
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	: Irfani Arija
NIM	: 210705110
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Judul	: perancangan sistem pemantauan kebisingan berbasis IoT dengan metode komunikasi Half-Duplex menggunakan ESP32 di masjid Fathun Qarib dan perpustakaan UIN Ar-Raniry

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memangditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh,
Yang Menyatakan

(Irfani Arija)

ABSTRAK

Nama : Irfani Arija
NIM : 210705110
Program Studi : Teknologi Informasi

Judul : Perancangan Sistem Pemantauan Kebisingan Berbasis Iot dengan Metode Komunikasi Half-Duplex Menggunakan ESP32 di Masjid Fathun Qarib dan Perpustakaan Uin Ar-Raniry

Tanggal Sidang :
Jumlah Halaman 49
Pembimbing I : Mulkan Fadhli,M.T
Pembimbing II : Muhammad Syamsu Rizal,M.T
Kata Kunci : Pemantauan Kebisingan, ESP32 Gateway, Half-Duplex

Kebisingan merupakan masalah yang berkembang seiring dengan pertumbuhan kepadatan penduduk, yang dapat mengganggu kenyamanan beraktivitas, baik saat beribadah maupun belajar. Penelitian ini bertujuan untuk merancang sistem pemantauan kebisingan berbasis Internet of Things (IoT) yang menggunakan ESP32 sebagai gateway dengan komunikasi Half-Duplex dan protokol HTTP. Sistem ini dirancang untuk menghubungkan sensor suara di dua lokasi, yaitu Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry, dan mengirimkan data kebisingan secara real-time ke platform ThingSpeak untuk visualisasi. Penggunaan komunikasi Half-Duplex memungkinkan peningkatan efisiensi transfer data. Hasil pengujian menunjukkan bahwa sistem ini dapat mengirimkan data kebisingan dengan latensi rendah dan stabilitas pengiriman yang baik pada kondisi jaringan yang stabil. Penelitian ini menawarkan solusi efisien untuk pemantauan kebisingan yang dapat dikembangkan lebih lanjut dengan perangkat dan teknologi yang lebih canggih, memberikan kontribusi dalam pengelolaan kebisingan di lingkungan publik.

Kata Kunci : Pemantauan Kebisingan, ESP32 Gateway, Half-Duplex

ABSTRACT

Name	: Irfani Arija
Student ID	210705110
Study Program	: *Information Technology*
Title	: *Design of Iot-Based Noise Monitoring System with Half-Duplex Communication Method Using ESP32 at Fathun Qarib Mosque and Uin Ar-Raniry Library*
Session Date	:
Thesis Thickness	*49*
Supervisor I	: Mulkan Fadhli, M.T
Supervisor II	: Muhammad Syamsu Rizal,M.T
Keywords	: *Noise Monitoring, ESP32 Gateway, Half-Duplex*

Noise is an issue that grows alongside population density, which can disturb the comfort of activities, both during worship and work. This research aims to design an Internet of Things (IoT)-based noise monitoring system using the ESP32 as a gateway with Half-Duplex communication and the HTTP protocol. The system is designed to connect sound sensors at two locations, namely Masjid Fathun Qarib and the UIN Ar-Raniry Library, and transmit noise data in real-time to the ThingSpeak platform for visualization. The use of Half-Duplex communication enhances data transfer efficiency. The test results show that the system can transmit noise data with low latency and stable transmission under stable network conditions. This research offers an efficient solution for noise monitoring that can be further developed with more advanced devices and technologies, contributing to noise management in public environments.

Keywords: Noise Monitoring, ESP32 Gateway, Half-Duplex

KATA PENGANTAR

Alhamdulillahi Rabbil ‘Alamin, puji dan syukur kita panjatkan ke hadirat Allah Subhanahu Wata’ala, Tuhan dengan kemahabesaran dan kemahakuasaan-Nya. Alhamdulillah atas segala nikmat dan rahmat-Nya yang terlimpahkan kepada kita sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Perancangan Sistem Pemantauan Kebisingan IoT dengan Half-Duplex Menggunakan ESP32 di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry.” Shalawat dan salam senantiasa kita sanjungkan kepada Rasulullah Shallallahu Alaihi Wasallam yang telah membawa kita dari alam jahiliyah ke alam Islamiyah seperti yang kita rasakan saat ini.

Tugas akhir ini disusun dan diselesaikan untuk memenuhi tugas akhir perkuliahan serta sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry. Selain itu, skripsi ini juga dibuat sebagai wujud implementasi dari ilmu yang diperoleh selama masa perkuliahan di Program Studi Teknologi Informasi.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis berharap dapat belajar lebih baik lagi dalam mengembangkan dan mengimplementasikan ilmu yang diperoleh ke depannya. Tugas akhir ini tentunya tidak terlepas dari tuntunan, bimbingan, dan masukan dari berbagai pihak. Oleh karena itu, sudah sepantasnya penulis dengan penuh hormat mengucapkan terima kasih dan mendoakan semoga Allah memberikan balasan terbaik kepada:

1. Orang tua luar biasa saya, yang selalu mendoakan dan memberi dukungan yang luar biasa selama saya menempuh pendidikan dan penulisan ini.
2. Kakak Rauza, kakak perempuan inspiratif saya yang telah banyak membantu baik dukungan moral maupun materi.
3. Bapak Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU., selaku Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh.

4. Ibu Malahayati, M.T dan Bapak Khairan AR, M.Kom selaku Ketua dan Sekretaris Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh, yang semangat dan motivasi beliau selalu menjadi hal yang spesial bagi saya.
5. Bapak Mulkan Fadhli, M.T., selaku pembimbing saya, yang dengan kemurahan hati dan ilmunya selalu mengiringi proses penelitian hingga penulisan tugas akhir ini.
6. Bapak Muhammad Syamsu Rizal,M.T., selaku pembimbing saya, yang dengan kemurahan hati dan bimbingannya, telah membantu mulai dari penelitian hingga penulisan tugas akhir ini.
7. Ibu Cut Ida Rahmadiana, S.Si., selaku Staf Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
8. Azizatul Maula, Ryan Abdul Azis, Huznullizam, dan Kismullah, mereka yang dalam penelitian ini memiliki andil besar dalam membantu penulis dalam melakukan pengetesan alat dan menemani selama penulisan tugas akhir ini.

Akhir kata, penulis menyadari bahwa tidak ada yang luput dari kesilapan. Apabila terdapat kesalahan, baik dalam penulisan maupun penyusunan tugas akhir ini, penulis memohon maaf yang sebesar-besarnya atas kekurangan yang ada. Penulis berharap semoga tugas akhir ini dapat bermanfaat bagi pembaca dan dijadikan referensi untuk pengembangan penelitian ke arah yang lebih baik. Kebenaran datangnya dari Allah, dan kesalahan berasal dari penulis sendiri. Semoga Allah SWT senantiasa melimpahkan nikmat dan karunia-Nya kepada kita semua.

Banda Aceh,10 November 2024

Penulis

Irfani Arija

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

I.1 Latar Belakang

Seiring dengan meningkatnya kepadatan penduduk di Indonesia yang mencapai 282,48 juta jiwa pada Juni 2024 (Direktorat Jenderal Kependudukan dan Pencatatan Sipil, 2024), masalah kebisingan semakin tidak terkendali, baik di luar maupun dalam gedung. Oleh karena itu, pengelolaan kebisingan perlu mendapatkan perhatian serius. Berdasarkan penelitian sebelumnya, kebisingan merupakan gangguan yang dapat dideteksi oleh telinga manusia (Andianingsari & Rahman, 2023). Kondisi ini mendorong perlunya solusi dengan pemanfaatan teknologi yang dapat mendukung terciptanya lingkungan yang nyaman, terutama di dalam gedung. Pemanfaatan teknologi IoT (*Internet of Things*) untuk pemantauan kebisingan merupakan langkah yang sejalan dengan perkembangan revolusi industri 4.0. Dalam hal ini, pemilihan metode komunikasi yang sesuai juga merupakan hal yang sangat penting diperhatikan.

Oleh karena itu, penelitian ini berjudul "Perancangan Sistem Pemantauan Kebisingan Berbasis IoT Menggunakan Metode Komunikasi Half-Duplex pada Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry." Setelah penulis merancang sistem pemantauan kebisingan, sejalan dengan penelitian yang dilakukan oleh Satriawan et al. (2020) mengenai pemantauan kebisingan berbasis IoT, yang menunjukkan bahwa penggunaan teknologi IoT untuk pemantauan kebisingan dapat meningkatkan efektivitas dalam pengelolaan kebisingan di lingkungan publik. Selain itu, penelitian oleh Nugroho dan Pratama (2018) mengenai komunikasi data pada sistem berbasis IoT juga menunjukkan bahwa metode komunikasi Half-Duplex dapat mengoptimalkan transfer data dan meningkatkan efisiensi sistem. Dengan demikian, metode komunikasi Half-Duplex diyakini dapat mengatasi kendala teknis yang ada dan memastikan pengiriman data yang efisien dalam sistem pemantauan kebisingan yang telah dirancang. Dalam sistem ini, setiap perangkat, baik sensor maupun ESP gateway, dapat secara bergantian mengirimkan atau menerima data, sehingga pendekatan Half-Duplex

mengurangi risiko tabrakan sinyal dan data yang umum terjadi dalam komunikasi Full-Duplex ketika data dikirim dan diterima secara bersamaan. Dengan demikian, penggunaan metode Half-Duplex dalam penelitian ini memungkinkan penciptaan sistem pemantauan kebisingan yang optimal dari segi konsumsi daya, stabilitas data, dan biaya operasional.

Dengan adanya sistem pemantauan berbasis IoT dengan metode komunikasi Half-Duplex ini, diharapkan dapat membantu menciptakan lingkungan gedung yang lebih sehat dan sesuai dengan standar Keputusan Menteri Negara Lingkungan Hidup Republik Indonesia No. KEP-48/MENLH/11/1996 tentang Baku Tingkat Kebisingan. Dengan demikian, penelitian ini diharapkan menjadi kontribusi signifikan dalam mendukung terciptanya ruang publik yang lebih nyaman dan terkendali.

I.2 Rumusan Masalah

Berdasarkan gambaran latar belakang di atas, rumusan masalah dalam penelitian ini adalah Bagaimana perancangan sistem pemantauan kebisingan berbasis IoT dengan metode komunikasi half-duplex dapat memungkinkan ESP32 berfungsi ganda sebagai pengirim data dan gateway dalam mengukur dan menampilkan data kebisingan secara real-time di ThingSpeak di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry?

I.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah di atas, tujuan penelitian ini adalah untuk merancang dan mengimplementasikan sistem pemantauan kebisingan berbasis IoT dengan metode Half-Duplex, yang memungkinkan ESP32 berfungsi ganda sebagai pengirim data dan gateway, guna mengukur dan menganalisis tingkat kebisingan secara real-time di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry.

I.4 Batasan Masalah

Agar pembahasan tetap relevan dengan judul dan latar belakang penelitian, penulis akan memfokuskan penelitian ini pada perancangan dan implementasi sistem pemantauan kebisingan berbasis IoT dengan metode Half-Duplex. Penelitian ini akan mengeksplorasi kemampuan ESP32 sebagai pengirim data dan gateway, serta memastikan sistem dapat beroperasi dengan baik untuk memantau kebisingan secara real-time di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry.

I.5 Manfaat Penelitian

Dari gambaran latar belakang, perumusan masalah, dan tujuan penelitian yang telah disampaikan, berikut adalah beberapa manfaat yang diharapkan dapat diperoleh dari penelitian ini:

1. Bagi penulis, penelitian ini diharapkan dapat memberikan kontribusi pada peningkatan ilmu pengetahuan, terutama dalam penerapan Internet of Things (IoT) dan metode Half-Duplex dalam sistem pemantauan kebisingan. Penelitian ini juga dapat menjadi referensi bagi penelitian di masa mendatang, khususnya yang berkaitan dengan pengembangan sistem berbasis IoT.

2. Bagi lingkungan di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry, penerapan sistem pemantauan kebisingan berbasis metode Half-Duplex diharapkan tidak hanya berhasil mengontrol tingkat kebisingan, tetapi juga membuka peluang untuk pengembangan lebih lanjut dalam pemantauan lingkungan yang nyaman untuk kegiatan ibadah dan belajar.

BAB II
LANDASAN TEORI

II.1 Relevansi Penelitian

Dalam konteks penelitian yang akan dilakukan, penulis memerlukan sumber referensi atau studi terkait yang sudah dilakukan sebelumnya. Hal ini dilakukan untuk menghindari plagiarisme atau duplikasi dalam penelitian ini, serta sebagai bahan perbandingan, pembelajaran, dan pedoman dalam penulisan penelitian ini. Selain itu, untuk melihat perbedaan antara penelitian yang penulis lakukan dengan yang terdahulu, hal ini dapat dilihat pada Tabel II.1.

Tabel II.1 Relevansi Penelitian

No	Peneliti/tahun	Judul penelitian	Kesimpulan penelitian
1	Jaka Prayudha,dkk. 2023	Implementasi Teknik Komunikasi Serial Half Duplex Pada Kendali Jarak Jauh Lampu Ruangan Rumah Berbasis Internet Of Things (IOT)	Teknik komunikasi serial half duplex dapat diimplementasikan ke dalam sistem kendali jarak jauh.

2	Andianingsari Diah,dkk.2023	Pemantauan kebisingan dengan menggunakan peta kebisingan program suffer 15 di pt. F	Penelitian ini untuk pemantauan kebisingan gambaran yang jelas tentang distribusi kebisingan di kawasan industri.
3	Wilani,dkk.2023	Rancang bangun sistem monitoring kebisingan pada ruangan dengan sensor suara gy-max4466 yang terintegrasi dengan platform IoT.	Studi ini menunjukkan bahwa sistem ini mampu mengukur dan memantau tingkat kebisingan di lingkungan tertutup dengan akurasi yang memadai.
4	Bayu Prasetio,dkk.2022	Sistem monitoring kebisingan berbasis internet of things	Pendekatan berbasis IoT yang memanfaatkan WSN dapat meningkatkan efisiensi dalam pemantauan kebisingan dan membantu penanganan masalah kebisingan dengan cara yang lebih terintegrasi.

II.2 Gangguan suara

Gangguan suara adalah bunyi yang tidak diinginkan dalam tingkat dan waktu tertentu yang dapat menimbulkan gangguan kesehatan manusia dan kenyamanan lingkungan. . Kebisingan yang terjadi secara terus menerus dapat menimbulkan gangguan kesehatan dan ketidaknyamanan(Mujayin Kholik, & Dimas Adji Krishna (2012).Kebisingan biasanya dihasilkan dari berbagai macam kegiatan yang melibatkan banyak hal, baik aktivitas lalu lintas hingga interaksi manusia yang tanpa sadar mengganggu dan memengaruhi tingkat kenyamanan, kesehatan, atau konsentrasi lingkungan sekitarnya.

II.3 Standar Level kebisingan suara di Indonesia

Di Indonesia, masalah kebisingan juga menjadi perhatian khusus dari pemerintah, di mana terdapat peraturan langsung oleh Kementerian Negara Lingkungan Hidup mengenai standar- standar kebisingan di suatu tempat di Indonesia. Menurut Keputusan Menteri Negara Lingkungan Hidup KEP-48/MENLH/11/1996, standar-standar tersebut dijelaskan pada Tabel II.2.

Tabel II.2 Daftar Peraturan dari Kementerian Lingkungan Hidup

BAKU KEBISINGAN KAWASAN/LINGKUNGAN KEGIATAN	TINGKAT KEBISINGAN (DB)
Perumahan	55
Perdagangan dan jasa	70
Perkantoran dan perdagangan	65
Ruang terbuka hijau	50
Industri	70

Pemerintah dan fasilitas umum	60
Rekreasi	70
Khusus: Bandara Stasiun kereta api Pelabuhan Cagar budaya	 70 60
Rumah sakit atau sejenisnya	55
Sekolah atau sejenisnya	55
Tempat ibadah atau sejenisnya	55
Keterangan: Disesuaikan dengan ketentuan Menteri Perhubungan.	

II.4 *Internet of Think* (IoT)

Adalah sebuah konsep di mana objek fisik, seperti sensor atau alat, terhubung ke internet dan saling berinteraksi tanpa harus berinteraksi langsung dengan manusia. IoT adalah perpaduan satu kesatuan sistem yang memakai jaringan internet selaku penghubung (Sari et al., 2024).

Tujuan utama IoT adalah untuk menghubungkan fisik dengan dunia digital. Dunia fisik diukur dengan alat atau sensor-sensor dan diterjemahkan menjadi data-data yang bisa dibaca komputer (Nahdi et al., 2021). Berikut Ini adalah komponen utama dari sistem Internet of Things:

- Perangkat Fisik: Perangkat fisik dapat berupa aktuator, sensor, atau perangkat apa pun yang terhubung ke internet, seperti kamera ,sensor kelembapan, atau perangkat cerdas lainnya.

- Sensor dan Aktuator: Suhu, kelembapan, gangguan, atau gerakan dapat dikumpulkan oleh sensor dalam IoT ini. Aktuator dapat menyalakan dan mematikan lampu atau bahkan dapat membuka pintu dan jendela sebagai respons terhadap data tersebut.

- *Platform dan Cloud* IoT: Hasil data yang dikumpulkan dari perangkat internet biasanya dikirim ke platform IoT atau cloud untuk diproses, dianalisis, dan disimpan. Pusat manajemen data seperti ini menawarkan alat untuk mengontrol perangkat, membuat keputusan otomatis, dan memvisualisasikan data.

- Perangkat & Gateway Edge: Beberapa perangkat Internet of Things beroperasi di tepi perangkat, tempat data diproses secara lokal sebelum dikirim ke cloud untuk diproses lebih lanjut.

- Konektivitas (Jaringan): Jaringan diperlukan untuk menghubungkan perangkat satu dengan yang lain, serta ke cloud dalam sistem IoT. Beberapa teknologi komunikasi yang digunakan di dalam jaringan ini di antaranya adalah Wi-Fi, Bluetooth, Zigbee, LoRa, dan jaringan seluler.

II.5 Mengenal Half-Duplex

Komunikasi Half-Duplex merupakan komunikasi dua perangkat yang dapat bertukar data dari kedua arah dalam satu kanal (Setiawan et al., 2018.) tetapi tidak pada saat waktu yang bersamaan (Prayudha et al., 2020). Satu perangkat harus menunggu gilirannya untuk menerima data dalam sistem Half-Duplex sementara perangkat lainnya mengirim. Agar aliran komunikasi terjadi secara bergantian, perangkat lainnya dapat mulai mengirim data setelah perangkat pertama selesai.

Radio dua arah dan walkie-talkie merupakan contoh umum komunikasi Half-Duplex (Ardianto Pranata et al., 2020), di mana pengguna harus mengatakan "over" untuk berpindah giliran. Dalam komunikasi berbasis jaringan, mode Half-Duplex juga sering dipakai, misalnya, dalam beberapa protokol jaringan lama yang mengharuskan perangkat untuk mengirim dan menerima data secara bergantian pada satu saluran. Sistem Half-Duplex sendiri dapat memaksimalkan penggunaan bandwidth dengan cara ini, terutama ketika komunikasi dua arah secara bersamaan tidak diperlukan.

Penggunaan saluran komunikasi yang sederhana dan efektif merupakan dua manfaat komunikasi Half-Duplex, terutama ketika kedua perangkat hanya perlu mengirim data secara bergantian. Oleh karena itu, Half-Duplex sangat cocok untuk pengaturan dengan bandwidth terbatas dan ketika komunikasi simultan tidak diperlukan.Dan amat cocok untuk komunikasi jarak jauh (Prayudha et al., 2020).

II.6 HTTP Protocol

HTTP (*HyperText Transfer Protocol)* merupakan protokol komunikasi yang digunakan untuk pertukaran dan pengiriman data di internet. HTTP adalah pondasi dasar dari setiap pertukaran data di Web (Subiksha & kirthic, 2019), yang memungkinkan komunikasi antara *client dan server* (Arifin et al., 2018).

Protokol komunikasi ini memungkinkan pengiriman dan penerimaan data melalui jaringan antara perangkat, termasuk komputer dan perangkat *Internet of Things*, serta server web. Klien (seperti browser web atau perangkat ESP32) mengirimkan permintaan ke server melalui HTTP, dan server membalas dengan informasi yang diminta. HTTP menggunakan beberapa metode permintaan, seperti GET, POST, PUT, dan DELETE, untuk mengelola jenis interaksi yang terjadi antara klien dan server:

Karena HTTP menggunakan TCP/IP sebagai protokol transportnya dan berjalan pada lapisan aplikasi dalam model OSI, HTTP sebenarnya tidak memiliki batasan jarak kerja langsung. Hal ini dikarenakan HTTP dapat bekerja pada jarak yang sangat jauh (seperti komunikasi antarnegara atau benua) maupun jarak sangat dekat (seperti jaringan lokal) selama koneksi jaringan internet tersedia dan dapat diakses.dengan demikian inilah salah satu alasan kenapa penulis memilih protocol komunikasi ini.

II.7 ThingSpeak

ThingSpeak adalah platform berbasis cloud (Ningsih et al., 2022) yang dirancang khusus untuk aplikasi Internet of Things (IoT). Platform berbasis cloud ini diciptakan untuk membantu pengembangan Internet of Things (IoT). Platform ini memungkinkan pengguna untuk mengumpulkan, melihat, dan mengevaluasi data dari berbagai sensor dan perangkat. Melalui penggunaan antarmuka berbasis web, ThingSpeak memungkinkan pengguna untuk membuat dasbor interaktif yang memfasilitasi pemantauan data secara real-time dan menghubungkan perangkat melalui berbagai protokol komunikasi, salah satunya ptotokol HTTP yang dipakai dalam penelitian ini.

Pemrograman logika berbasis aturan dan integrasi dengan alat analisis data lainnya adalah dua fitur utama ThingSpeak. Berkat fitur ini, pengguna dapat dengan mudah melakukan analisis mendalam terhadap data yang dikumpulkan. ThingSpeak adalah platform serbaguna untuk berbagai aplikasi Internet of Things karena dapat menampilkan visualisasi yang sesuai dengan kebutuhan pengguna dan mendukung pengambilan data dari berbagai masukan (Khan et al., 2019).

II.8 Arduino IDE

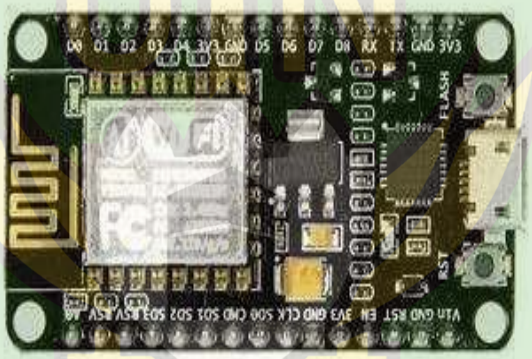
Arduino merupakan sebuah aplikasi yang berguna untuk membuat kode pemograman lalu meng-upload ke board yang dipilih (Mahanin Tyas et al., 2023),Arduino adalah aplikasi lintas platform yang dibuat dalam Java. Bahasa pemrograman Processing dibuat untuk memperkenalkan pemrograman kepada para seniman dan pemula di dunia IoT yang tidak terbiasa membuat perangkat lunak. Aturan kode khusus digunakan untuk mendukung pengaturan bahasa pemrograman C dan C++, dan Arduino IDE digunakan untuk menulis program yang disebut sketsa.Kesederhanaan dalam penggunaan ini merupakan salah satu nilai lebih yang ada dalam aplikasi ini. Fitur utamanya mencakup kemampuan untuk memprogram dan memuat program secara langsung ke papan Arduino.

II.9 Mikrokontroler NodeMCU ESP32

Mikrokontroler adalah sebuah chip IC (*Integrated Circuit*) dalam mini komputer yang dirancang untuk menjalankan perintah atau instruksi tertentu. Mikrokontroler ini terdiri dari beberapa inti pemrosesan (CPU), memori (RAM dan ROM), serta perangkat input dan output yang dijalankan melalui kode program. Pada penelitian ini digunakan Mikrokontroler NodeMCU ESP32, yaitu papan yang dirancang untuk membuat proyek pengembangan elektronik, yang menggabungkan Mikrokontroler ESP-32 dengan fitur tambahan seperti akses WiFi dan konverter USB to serial, memungkinkan pemrograman langsung menggunakan kabel USB. Cara kerjanya melibatkan koneksi NodeMCU ke komputer melalui USB, di mana pengguna

dapat mengunggah kode menggunakan Arduino IDE. Setelah kode diunggah, NodeMCU dapat berfungsi secara mandiri untuk mengontrol perangkat atau berkomunikasi dengan jaringan WiFi sesuai dengan instruksi program. Papan ini berfungsi sebagai media komunikasi antara port serial dan USB untuk pemrograman dan komunikasi (Lukita Sari & Lailia Maulidina, 2022).

Dengan kemampuannya untuk beroperasi pada frekuensi clock yang bervariasi, biasanya dalam kisaran 80 MHz hingga 240 MHz, prosesor ini menawarkan fleksibilitas untuk berbagai aplikasi pemrosesan. Penyimpanan data dan program yang lebih besar dimungkinkan dengan NodeMCU ESP32 berkat RAM 520 KB dan memori Flash, yang bervariasi antara 4 MB dan 16 MB tergantung pada versi papan.Mikrokontroler NodeMCU ESP32 dapat dilihat pada Gambar II.1.



Gambar II.1 NodeMCU ESP32

Sumber: https://www.aiophotoz.com/

II.10 KY-037 Sound Sensor

Sensor suara ini digunakan untuk mendeteksi tingkat suara di lingkungan sekitarnya. Cara kerja sensor ini adalah dengan mengubah gelombang suara menjadi sinyal listrik, yang dapat digunakan untuk berbagai aplikasi atau proyek pengukuran suara(Nabawi & Badarudin, 2024). KY-037 Sound sensor Module dapat dilihat pada Gambar II.2.

Gambar II.2 Sound Detection Module KY-037

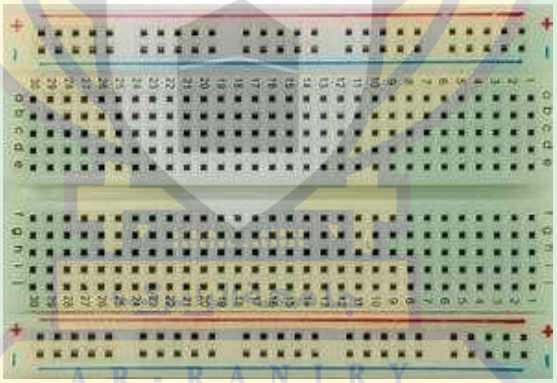
Sumber: http://indomaker.com/

- Fitur dan Spesifikasi:
- Sensor deteksi kebisingan Suara
- Berada di tegangan : 3.3V hingga 5V DC
- Output Analog: Menghasilkan sinyal analog yang sebanding dengan tingkat suara.
- Output Digital: Menghasilkan sinyal HIGH/LOW berdasarkan ambang batas suara.
- Arus pengoperasian: 4~5 mA
- Jarak Deteksi: Tergantung intensitas (umumnya20Hz).

II.11 Breadboard

Breadboard adalah dasar konstruksi sebuah sirkuit purwarupa dari suatu rangkaian elektronik (Tri Putra & Hamka Air Tawar, 2021). Rangkaian elektronik yang dirakit tanpa menyolder komponen menggunakan papan ini. Biasanya, papan ini terdiri dari papan plastik dengan baris dan kolom kontak logam yang terhubung di bawah permukaannya. Penempatan komponen elektronik yang sementara dan fleksibel memudahkan pengujian dan modifikasi rangkaian sebelum papan sirkuit cetak permanen dibuat, karena tidak memerlukan penyolderan.

Komponen elektronik yang dimasukkan ke dalam lubang pada papan agar dapat digunakan untuk membangun berbagai rangkaian. Dalam pengembangan dan eksperimen IoT, papan tempat memotong roti ini sangat berguna, memungkinkan pengguna untuk membangun dan memodifikasi rangkaian dengan cepat, serta mengidentifikasi dan menyelesaikan masalah sebelum menetapkan cetak biru akhir. Gambar Breadboard dapat dilihat pada Gambar II.3.



Gambar II.3 Breadboard

Sumber: https://www.iot-store.com.au/

II.12 Kabel jumper

Kabel jumper adalah kabel elektrik yang mempunyai pin konektor di tiap ujungnya, yang memungkinkan untuk menghubungkan dua komponen yang melibatkan Arduino tanpa memerlukan solder. Intinya, kegunaan kabel jumper ini adalah sebagai konduktor listrik untuk menyambungkan rangkaian listrik. Ujung kabel diklasifikasikan menjadi dua yaitu konektor jantan *(male connector)* dan konektor betina *(female connector).*

Kabel jumper berkerja untuk menghantarkan arus listrik dari satu komponen ke komponen lainnya yang dihubungkan di breadboard tanpa memerlukan solder (Theodorus S. Kalengkongan et al., 2018). Gambar kabel jumper female to female, male to female, dan male to male dapat dilihat di Gambar II.4.

Gambar II.4 kabel Jumper

Sumber: https://id.aliexpress.com/

BAB III
METODE PENELITIAN

III.1 Waktu dan Tempat Penelitian

Penelitian yang berjudul 'Perancangan Sistem Pemantauan Kebisingan berbasiss IoT dengan metode komunikasi Half-Duplex Menggunakan ESP32 di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry' ini dilaksanakan dari bulan Agustus 2024 hingga November 2024.

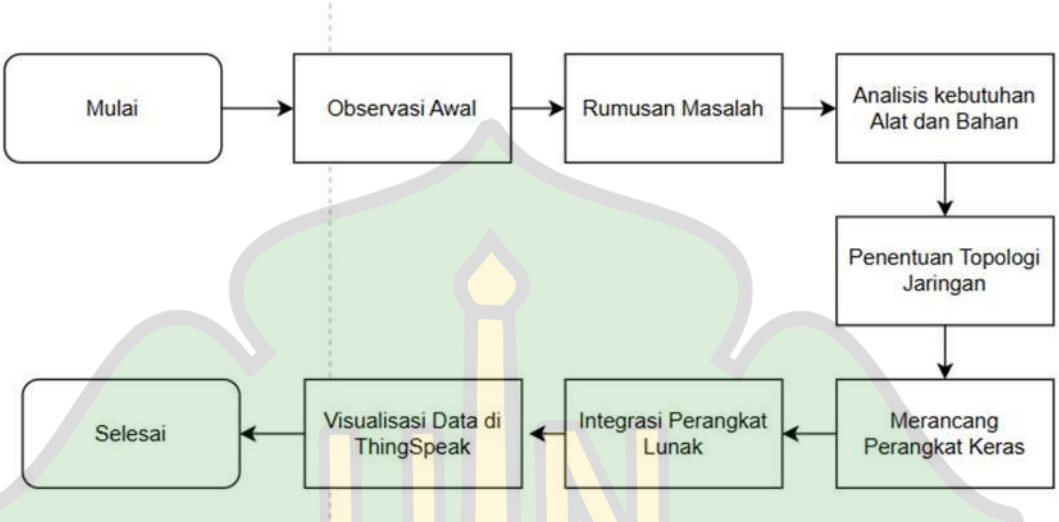
III.2 Kerangka Berfikir

Untuk penelitian Perancangan Sistem Pemantauan Kebisingan berbasis IoT dengan metode komunikasi Half-Duplex Menggunakan ESP32 di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry ini, kerangka berpikir disajikan pada Gambar III.1

Gambar III.1 Kerangka Berfikir

III.3 Alur Penelitian

Alur penelitian ini adalah langkah-langkah atau proses dimana penelitian ini dimulai hingga selesai ini dituangkan pada Gambar III.2.



Gambar III.2 Diagram Penelitian

III.3.1 Observasi awal

Pada tahap ini, penulis melakukan kajian terhadap penelitian terdahulu yang relevan serta melakukan survei lapangan langsung ke beberapa fasilitas publik. Data dikumpulkan sesuai dengan studi kasus yang sedang dibahas, baik melalui observasi langsung maupun metode tidak langsung.

III.3.2 Rumusan masalah

Rumusan masalah adalah pernyataan atau pertanyaan yang merumuskan isu atau tantangan yang ingin dipecahkan melalui penelitian perancangan sistem pemantauan kebisingan ini. Rumusan masalah di penelitian ini adalah: “Bagaimana perancangan sistem pemantauan kebisingan berbasis IoT dengan metode Half-Duplex dapat memungkinkan ESP32 berfungsi ganda sebagai pengirim data dan gateway dalam mengukur data kebisingan di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry?”

III.3.3 Analisis Kebutuhan Alat dan Bahan

Penulis melakukan analisis terhadap alat dan bahan yang akan digunakan serta menyesuaikan alat tersebut dengan fungsinya untuk melancarkan penelitian ini. Dalam hal ini, alat-alat dan bahan yang di gunakan di antaranya dapat dilihat pada Tabel III.1 Alat dan bahan.

Tabel III.1 Alat dan bahan

No	Kebutuhan Perangkat keras	Kebutuhan perangkat lunak
1	ESP32	Arduino IDE
2	KY-037Sound Sensor	ThingSpeak
3	Power Supply	HTTP Protocol
4	Breadbord	
5	Kabel jumper	
6	Box Project	

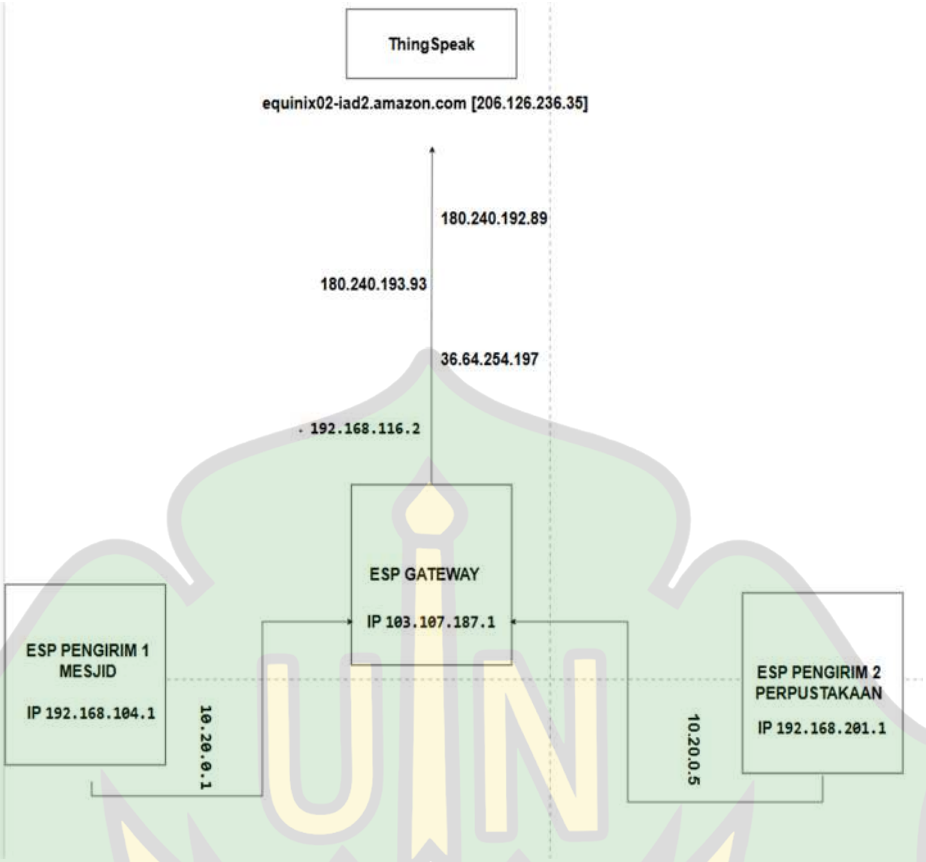
III.3.4 Penentuan Topologi Jaringan

Pada tahap ini, peneliti melakukan analisis untuk menentukan topologi jaringan yang akan digunakan, dimulai dari penyesuaian lokasi sistem yang akan pasang, penempatan Mikrokontroler sesuai dengan jarak jangkauannya, serta pemilihan jenis jaringan yang akan digunakan,adalah langkah langkah yang di pakai peneliti akhirnya memilih jenis topologi Star untuk di gunakan.

III.3.4.1 Topologi Star

Topologi Star memungkinkan semua perangkat atau node dalam jaringan dengan topologi bintang terhubung langsung ke hub pusat, seperti switch, router, atau gateway. Dalam penelitian ini, satu ESP32 berfungsi sebagai gateway melalui jaringan Wi-Fi kampus, dan dua sensor noise terhubung dengannya di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry Melalui saluran komunikasinya yang terpisah, setiap sensor noise mengirimkan data langsung ke ESP32 gateway, yang kemudian meneruskannya ke ThingSpeak. Dalam implementasi jaringan kontemporer, topologi ini sangat diinginkan karena isolasi kesalahannya yang lebih baik (Hussain et al., 2021).

Dalam jenis topologi ini, semua perangkat atau node memiliki jalur koneksi sendiri ke titik pusat, sehingga data yang dikirim dari satu perangkat harus melewati titik pusat sebelum mencapai perangkat tujuan. Misalnya, dalam jaringan Wi-Fi, router bertindak sebagai titik pusat yang menghubungkan semua perangkat, seperti komputer atau sensor, ke jaringan Wi-Fi *(Wireless Fidelity*). Router juga menghubungkan semua perangkat. Topologi bintang memberikan kemudahan pengelolaan dan pemeliharaan, serta meningkatkan kinerja jaringan (Khan et al., 2020). Penerapan topologi dalam penelitian ini dapat dilihat pada Gambar III.3.

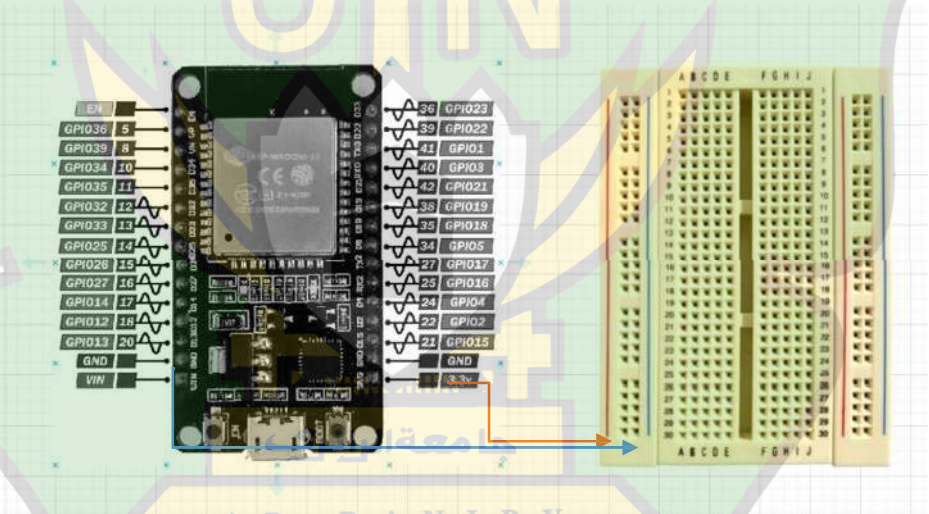


Gambar III.3 Rancangan Topologi Star

III.3.5 Perancangan perangkat keras

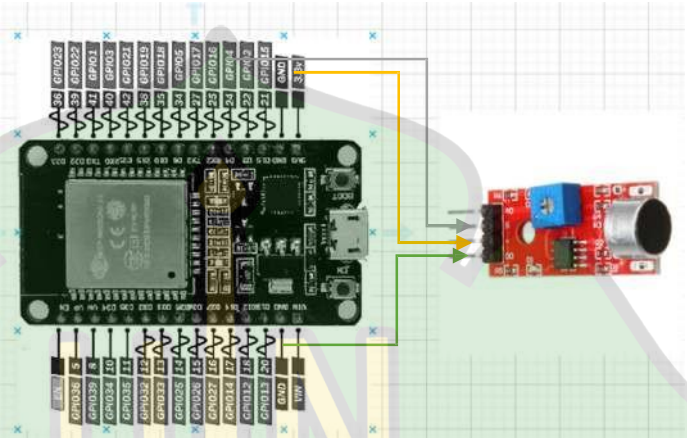
Perancangan alat ini mencakup keseluruhan proses pembuatan perangkat keras yang digunakan dalam penelitian ini, mulai dari pemasangan kabel antar ESP32 dengan sensor kebisingan hingga integrasi komponen perangkat keras lainnya untuk memastikan bahwa semua komponen berfungsi secara optimal. Selain itu, proses ini melibatkan pengujian dan penyesuaian komponen untuk memastikan bahwa perangkat keras dapat menjalankan fungsinya sesuai dengan tujuan penelitian. Langkah perancangan perangkat keras penelitian ini adalah sebagai berikut:

1. Persiapan menghubungkan ESP-32 dengan breadboard meliputi proses menghubungkan pin daya VCC pada ESP ke jalur daya (+) di breadboard dan menghubungkan pin GND pada ESP ke jalur ground (-) di breadboard.Dapat dilihat pada Gambar III.4.



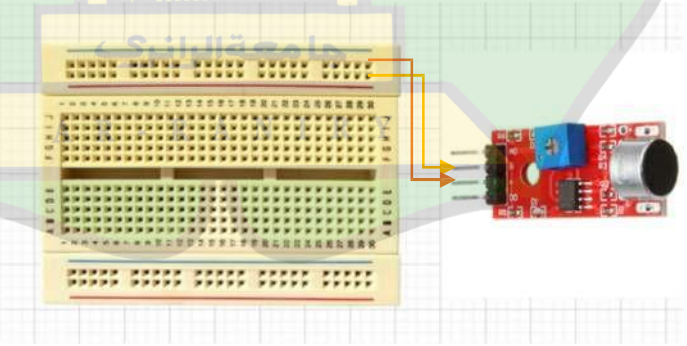
Gambar III.4 Menghubungkan ESP-32 dengan Breadboar

2. Pemasangan sensor KY-037 ke ESP-32 adalah proses menghubungkan sensor suara KY-037 dengan ESP-32 dimana hal ini dimaksudkan untuk membuat dua perangkat tersebut bisa terhubung melalui Analog Output dari sensor ke pin D34 pada ESP-32.Dapat dilihat pada Gambar III.5.



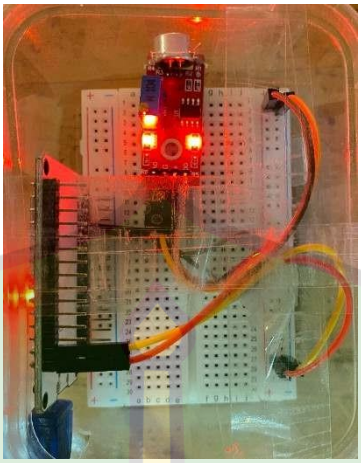
Gambar III.5 Pemasangan sensor suara KY-037 ke ESP-32

3. Menghubungkan sensor suara KY-037 ke Breadboard untuk mendapakan pembagian arus untuk dapat menyediakan energi yang diperlukan untuk mengoperasikan sensor.Proses penyambungan dua perangkat tersebut dapat dilihat pada Gambar III.6.



Gambar III.6 Menghubungkan sensor suara KY-037 ke Breadboard

4. Alat pemantauan kebisingan selesai dan siap digunakan setelah melalui beberapa tahap rancangan. Alat pemantauan kebisingan ini dapat dilihat seperti pada Gambar III.7.



Gambar III.7 Alat pemantauan kebisingan

III.3.6 Integrasi Perangkat Lunak

Dalam pengembangan penelitian ini, sangat dibutuhkan integrasi yang sesuai antara perangkat keras dengan pengembangan perangkat lunak untuk mendukung sistem pemantauan kebisingan agar berjalan sesuai dengan rencana. Perangkat lunak di sini menggunakan Arduino IDE, yaitu perangkat di mana kode pemrograman ini dibuat yang akan menghidupkan fungsi perangkat keras lalu mengelola data dari sensor. Pembuatan kode pemrograman untuk Mikrokontroler, seperti ESP32, merupakan langkah awal dalam penerapan Arduino IDE dalam studi pemantauan kebisingan Internet of Things ini.

Kode tersebut menentukan konfigurasi pin antara Mikrokontroler dan sensor suara serta mengatur komunikasi keduanya. Dengan instruksi yang jelas, Mikrokontroler secara berkala mengambil data dari sensor suara, yang memungkinkan pengukuran tingkat kebisingan di lokasi seperti Masjid Fathun Qarib dan Perpustakaan UIN Ar-RanirySetelah data terkumpul, logika pemrosesan yang mengubah nilai analog menjadi digital dapat diimplementasikan menggunakan Arduino IDE. Setelah itu, data ini dikirim ke platform ThingSpeak menggunakan protokol komunikasi yang sesuai.seperti HTTP.

a. Aktivasi Sensor suara KY-037 dan Kalibrasi

1. Membaca nilai dari sensor suara

   Bagian ini membaca nilai dari sensor suara yang terhubung ke pin analog 34: analogRead(noiseSensorPin): Fungsi ini mengembalikan nilai antara 0 hingga 4095, yang merupakan nilai ADC (*Analog-to-Digital Converter*) dari sensor. Ini merupakan pembacaan langsung dari sensor.

```
int noiseLevel = analogRead(noiseSensorPin); // Membaca nilai dari sensor suara
```

2. Mengonversi Nilai ADC ke Voltase

   noiseLevel / 4095.0): Bagian ini membagi nilai ADC dengan 4095 untuk mendapatkan proporsi dari nilai maksimum ADC.

   3.3: Ini mengalikan proporsi tersebut dengan 3.3 volt (tegangan referensi). Jadi, rumus ini mengonversi nilai ADC menjadi voltase yang sesuai.

```
float voltage = (noiseLevel / 4095.0) * 3.3; // Konversi ke voltase
```

3. Mengonversi Voltase ke Desibel

   Voltage 50.0 + 30.0: Di sini, menggunakan rumus konversi yang disederhanakan. voltage * 50.0: Mengalikan voltase dengan faktor pengali 50.0, yang mungkin ditentukan berdasarkan karakteristik sensor atau penguatan.

   + 30.0: Menambahkan offset 30 dB untuk menyesuaikan hasil agar lebih akurat. Penyesuaian ini dapat digunakan untuk kalibrasi, tergantung pada spesifikasi sensor dan kebutuhan aplikasi Anda.

```
float dB = voltage * 50.0 + 30.0; // Konversi ke desibel dan menambahkan 30 dB
```

4. Kondisi Pengiriman Pengaturan Ambang Batas Kebisingan

   Kemudian, kode ini hanya akan mengirimkan data jika nilai dB berada dalam rentang tertentu (di atas 10 dB dan di bawah 120 dB).

```
if (dB > 10 && dB < 120) {
  if (client.connect(gateway_ip, gateway_port)) { // Menghubungkan ke ESP gateway
    // Mengirim data kebisingan
  }
}
```

- Proses Kalibrasi Sensor

  Kalibrasi sensor suara yang digunakan di masing-masing ESP Pengirim dilakukan untuk memastikan akurasi dan keandalan pengukuran kebisingan. Proses kalibrasi ini penting untuk menjamin bahwa data yang dihasilkan mencerminkan tingkat kebisingan yang sebenarnya di lingkungan yang diamati.

5. Rumus untuk Menghitung Voltase dari Nilai ADC.

$$Voltage = (noise\ level\ \div 4095.0) \times 3.3$$

- noiseLevel adalah nilai yang dibaca dari sensor suara (dalam rentang 0 hingga 4095).
- 3.3 adalah tegangan referensi yang digunakan oleh ESP32.

6. Rumus untuk Menghitung Desibel (dB) dari Voltase.

$$dB = Voltage\ \times 50.0 + 30.0$$

- voltage adalah hasil konversi voltase dari langkah sebelumnya.
- 50.0 adalah faktor pengali yang mungkin digunakan untuk mengonversi voltase menjadi desibel.
- 30.0 adalah dB yang di tambahkan untuk kalibrasi hasil menjadi lebih akurat mengingat kekurangan sensor KY-037 yang tidak terlalu resposif terhadap suara yang kecil.

a. Berikut adalah tabel yang menunjukkan perbandingan antara hasil kalibrasi pengukuran sensor suara dan standar desibel yang telah ditentukan, dapat dilihat pada Tabel hasil kalibrasi sensor III.2.

Tabel hasil kalibrasi sensor III.2

Pengetesan kalibrasi	Sensor KY-037	Sound Level Meter	Selisih dB
1	68 dB	71 dB	3 dB
2	67 dB	69 dB	2 dB
3	64 dB	67 dB	3 dB
4	66 dB	68 dB	2 dB
5	63 dB	66 dB	3 dB

Tabel di atas menunjukkan hasil kalibrasi yang telah dilakukan dalam penelitian ini. Selisih antara hasil sensor suara dan Sound level Meter mencerminkan tingkat akurasi pengukuran sensor. Hasil ini menegaskan bahwa meskipun sensor memberikan hasil yang cukup baik, ada beberapa kendala yang dihadapi, terutama dalam perbedaan waktu yang di gunakan sensor untuk mengeluarkan output ,dimana sensor perlu waktu beberapa detik sesuai delay dalam kode untuk memproses data kebisingan sedangkan Aplikasi Sound Level Meter tidak memerlukan waktu yang lama dikarenakan bersifat offline. Dan kondisi lingkungan yang bising secara tiba. Oleh karena itu, sangat perlunya pemakaian sensor yang berkualitas lebih baik untuk menampilkan hasil kalibrasi yang lebih baik.

b. Konfigurasi kode ESP32 Gateway untuk Menjadi Mode Receiver

Proses konfigurasi kode ESP32 Gateway untuk berfungsi sebagai mode *Receiver* melibatkan pengaturan ESP32 untuk mendengarkan dan menerima data dari perangkat pengirim, seperti sensor kebisingan yang terhubung ke ESP32 lain. Dalam mode ini, ESP32 akan menerima data melalui permintaan HTTP GET yang dikirim dari perangkat pengirim. Kode di dalam ESP32 Gateway harus mencakup pengaturan untuk mendengarkan pada port yang telah ditentukan (umumnya port 80), serta memproses dan menyimpan data yang diterima. Selain itu, ESP32 juga perlu mengatur koneksi Wi-Fi untuk memastikan bahwa perangkat dapat terhubung ke jaringan dan siap untuk menerima data dari sensor kebisingan.

1. Inisialisasi Koneksi Wi-Fi

Bagian ini menginisialisasi koneksi Wi-Fi ESP32 dan membuat objek server yang akan mendengarkan permintaan HTTP pada port 80. Koneksi ke jaringan Wi-Fi dilakukan dengan menggunakan SSID WIFI_RANIRY tanpa password. Setelah terhubung, alamat IP lokal dari ESP32 dicetak ke Serial Monitor untuk referensi.

```
#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "WIFI_RANIRY"; // SSID jaringan Wi-Fi
const char* password = ""; // Tidak ada password

WebServer server(80); // Deklarasi objek server pada port 80
float noiseMosque = 0.0; // Variabel untuk menyimpan nilai kebisingan dari Masjid
float noiseLibrary = 0.0; // Variabel untuk menyimpan nilai kebisingan dari Perpustaka

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password); // Memulai koneksi Wi-Fi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print("."); // Menunggu hingga terhubung ke Wi-Fi
  }

  Serial.println();
  Serial.print("Terhubung ke Wi-Fi! IP Address: ");
  Serial.println(WiFi.localIP()); // Menampilkan alamat IP lokal
}
```

2. Mengatur Endpoint dan Memulai Server

Setelah terhubung ke Wi-Fi, bagian ini mengatur endpoint yang akan digunakan untuk menerima data. Endpoint yang digunakan adalah /data, di mana ESP32 akan mendengarkan permintaan dari pengirim. Setelah pengaturan selesai, server mulai mendengarkan permintaan dari klien.

```
void setup() {
  // Kode sebelumnya ...

  server.on("/data", HTTP_GET, handleData); // Menangani permi
  server.begin(); // Memulai server
}

void loop() {
  server.handleClient(); // Menangani klien yang terhubung
}
```

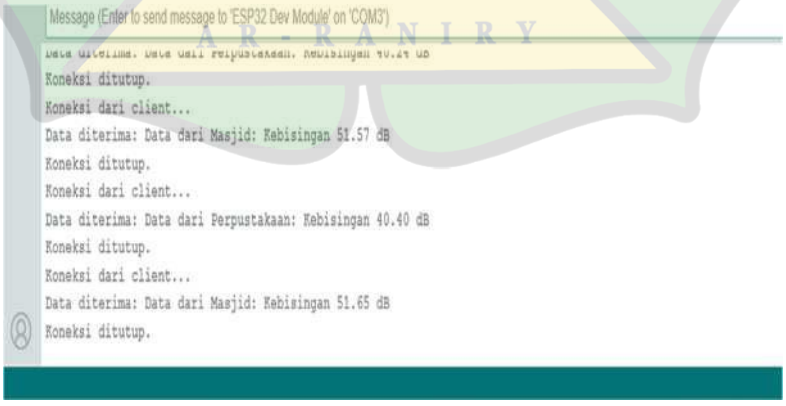
3. Pengolahan Data dari *Client*

Bagian ini mendefinisikan fungsi handleData() yang akan dipanggil ketika ada permintaan GET ke endpoint /data. Fungsi ini memeriksa apakah argumen noise ada dan menyimpan nilai kebisingan yang diterima berdasarkan alamat IP pengirim. Setelah memproses data, respons dikirim kembali ke pengirim untuk mengonfirmasi bahwa data telah diterima.

```
void handleData() {
  IPAddress clientIP = server.client().remoteIP(); // Mendapatkan alamat IP klien
  Serial.print("Alamat IP klien: ");
  Serial.println(clientIP); // Menampilkan alamat IP klien

  if (server.hasArg("noise")) {
    String noiseValue = server.arg("noise"); // Mengambil nilai dari argumen "noise"
    noiseValue.replace(",", "."); // Mengganti ',' dengan '.' jika ada
    float noise = noiseValue.toFloat(); // Mengonversi nilai ke float
    noise += 30.0; // Menambahkan 30 dB ke nilai kebisingan

  // Cek dari mana data datang dan simpan nilainya
  if (clientIP == IPAddress(192, 168, 201, 212)) { // Alamat IP pengirim Masjid
    noiseMosque = noise;
    Serial.print("Data diterima dari Masjid: ");
    Serial.println(noiseMosque);
  } else if (clientIP == IPAddress(192, 168, 201, 153)) { // Alamat IP pengirim
    noiseLibrary = noise;
    Serial.print("Data diterima dari Perpustakaan: ");
    Serial.println(noiseLibrary);
  } else {
    Serial.println("Data diterima dari sumber yang tidak dikenal.");
  }

  // Mengirimkan respons kembali ke pengirim
  server.send(200, "text/plain", "Data diterima"); // Mengirim respons sukses
} else {
  server.send(400, "text/plain", "Parameter tidak valid"); // Mengirim respons e
```

4. Hasil Output ESP untuk menjadi Mode *Receiver*

Hasil Output esp sebagai mode penerima data bisa dilihat pada Gambar III.8.



Gambar III.8 Output ESP untuk menjadi Mode *Receiver*

c. Pemograman dan Pengaturan ESP32 Pengirim 1 di Masjid Fathun Qarib

Dibagian ini penulis melakukan persiapan dan pemograman untuk ESP Pengirim 1 di Mesjid.

1. Implementasi Kode ESP Pengirim di Masjid Fathun Qarib

Kode program yang diterapkan pada ESP32 yang berfungsi sebagai pengirim data kebisingan dari Masjid Fathun Qarib ke ESP32 gateway. Kode ini menggunakan library Wi-Fi untuk menghubungkan ESP ke jaringan Wi-Fi lokal, melakukan pembacaan data dari sensor suara, dan mengirimkan data kebisingan ke ESP gateway pada interval tertentu.

2. Inisialisasi Konfigurasi Wi-Fi

Kode berikut mengatur konfigurasi koneksi Wi-Fi dan variabel yang dibutuhkan dalam pengiriman data ke ESP32 gateway.

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "Server";                // SSID Wi-Fi
const char* password = "pengurus02";        // Password Wi-Fi
const char* gatewayIP = "192.168.201.153";  // Alamat IP ESP gateway

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);  // Memulai koneksi Wi-Fi
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nTerhubung ke Wi-Fi!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
}
```

Di bagian ini, kita mengimpor pustaka yang diperlukan dan mendeklarasikan variabel untuk SSID, password, dan alamat IP gateway. Setelah itu, kita mulai koneksi Wi-Fi dalam fungsi setup() dan menunggu hingga koneksi berhasil. Ketika terhubung, alamat IP lokal akan ditampilkan di Serial Monitor.

3. Pengukuran tingkat kebisingan

Bagian ini membaca nilai kebisingan dari sensor suara yang terhubung ke pin analog 34 pada ESP32. Nilai analog yang terbaca dikonversi ke voltase, kemudian dikalibrasi menjadi desibel (dB) untuk memberikan hasil yang lebih akurat. Setelah itu, nilai kebisingan dalam desibel ditampilkan di Serial Monitor.

```
int noisePin = 34; // Pin analog sensor suara

// Fungsi untuk membaca level kebisingan dan mengonversinya ke dB
float readNoiseLevel() {
  int noiseLevel = analogRead(noisePin);        // Membaca nilai ADC dari sensor
  float voltage = (noiseLevel / 4095.0) * 3.3;  // Konversi ke voltase
  return (voltage * 50.0) + 30.0;               // Kalibrasi ke dB
}

void loop() {
  float dbNoiseValue = readNoiseLevel();  // Baca kebisingan dalam dB
  Serial.print("DB KEBISINGAN FATHUN QARIB: ");
  Serial.println(dbNoiseValue);
  sendNoiseData(dbNoiseValue);            // Kirim data ke gateway
  delay(10000);                           // Jeda 10 detik
}
```

4. Pengiriman Data ke Gateway

Pada bagian ini, data kebisingan dalam dB dikirim ke gateway (dengan alamat IP 192.168.201.153) menggunakan protokol HTTP. Jika koneksi berhasil, kode respons HTTP dari server ditampilkan di Serial Monitor. Proses ini diulang setiap 10 detik.

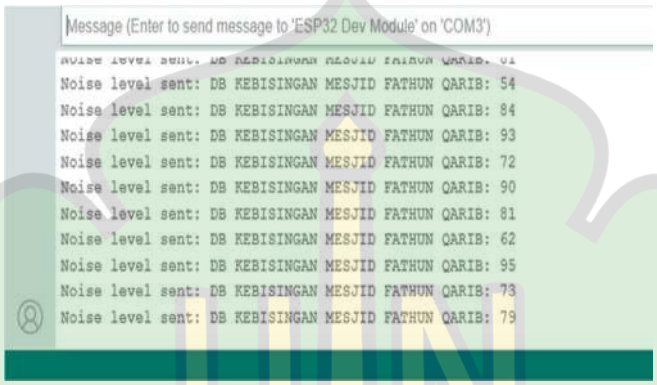
```
void sendNoiseData(float noise) {
  if (WiFi.status() == WL_CONNECTED) {      // Pastikan Wi-Fi terhubung
    HTTPClient http;
    String url = "http://" + String(gatewayIP) + "/data?noise=" + String(noise);
    http.begin(url);                        // Menginisiasi HTTP request ke gateway

    int httpResponseCode = http.GET();      // Mengirim data dengan metode GET
    if (httpResponseCode > 0) {
      Serial.print("Data terkirim, kode respons: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error pengiriman data, kode: ");
      Serial.println(httpResponseCode);
    }
    http.end(); // Mengakhiri koneksi HTTP
  } else {
    Serial.println("WiFi tidak terhubung.");
```

Di bagian ini, kode memeriksa status koneksi Wi-Fi sebelum memulai pengiriman data. Jika terhubung, objek HTTPClient digunakan untuk membuat permintaan HTTP GET dengan URL yang telah dibentuk. Kode juga memeriksa kode respons dari server

dan mencetak hasilnya di Serial Monitor. Jika koneksi tidak terhubung, kode akan memberikan pesan kesalahan.

5. Hasil Output ESP pengirim Mesjid Fathun Qarib

Setelah menulis kode untuk esp pengirim mesjid, lakukan run pada Arduino IDE untuk melihat hasil print Serial Monitor seperti Gambar III.9.



Gambar III.9 Output ESP pengirim Mesjid Fathun Qarib

d. Pemrograman dan Pengaturan ESP32 Pengirim 2 di Perpustakaan

Kode program untuk ESP32 yang berfungsi sebagai pengirim data kebisingan dari Perpustakaan UIN Ar-Raniry ke ESP32 gateway. Kode ini menggunakan library Wi-Fi untuk koneksi jaringan, pembacaan data dari sensor suara, dan pengiriman data kebisingan ke ESP gateway pada rentang waktu tertentu.

1. Inisialisasi Koneksi Wi-Fi

Pada bagian inisialisasi, kode menetapkan konfigurasi Wi-Fi dan variabel yang dibutuhkan dalam proses pengiriman data ke gateway.

```
#include <WiFi.h>
#include <HTTPClient.h>

// Konfigurasi Wi-Fi
const char* ssid = "WIFi-UINAR";                // SSID jaringan Wi-Fi
const char* password = "";                      // Kosongkan untuk tanpa

const char* gatewayIP = "192.168.201.153";     // Alamat IP ESP gateway d
int noisePin = 34;                             // Pin analog sensor suara

void setup() {
  Serial.begin(115200);
```

```
  // Koneksi Wi-Fi
  WiFi.begin(ssid, password); // Memulai koneksi Wi-Fi
  while (WiFi.status() != WL_CONNECTED) { // Tunggu hingga terhubung
    delay(500);
    Serial.print("."); // Menampilkan titik saat menunggu koneksi
  }
  Serial.println("\nTerhubung ke Wi-Fi!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP()); // Menampilkan alamat IP yang didapat
}
```

Di bagian ini, kita mengimpor pustaka yang diperlukan dan mendeklarasikan variabel untuk SSID, password, dan alamat IP gateway. Setelah itu, kita mulai koneksi Wi-Fi dalam fungsi setup() dan menunggu hingga koneksi berhasil. Ketika terhubung, alamat IP lokal akan ditampilkan di Serial Monitor.

2. Pengukuran dan Pengolahan Data Kebisingan

Kode ini berfungsi untuk membaca data dari sensor kebisingan yang terhubung dan mengkonversinya ke dalam satuan desibel (dB). Fungsi readNoiseLevel() digunakan untuk melakukan pengukuran ini.

```
void loop() {
  float dbNoiseValue = readNoiseLevel();  // Membaca nilai kebisingan dalam dB
  Serial.print("DB KEBISINGAN PERPUSTAKAAN: "); // Menampilkan pesan di Serial Monitor
  Serial.println(dbNoiseValue); // Menampilkan nilai kebisingan yang terbaca

  sendNoiseData(dbNoiseValue); // Mengirim data ke gateway
  delay(10000); // Jeda 10 detik
}

// Fungsi untuk membaca level kebisingan dan mengonversinya ke dB
float readNoiseLevel() {
  int noiseLevel = analogRead(noisePin); // Baca nilai ADC dari sensor
  float voltage = (noiseLevel / 4095.0) * 3.3; // Konversi ke voltase
  return (voltage * 50.0) + 30.0; // Konversi ke dB dengan kalibrasi
}
```

3. Bagian Pengiriman Data ke Gateway

Bagian ini mengatur pengiriman data kebisingan yang telah diproses ke gateway melalui HTTP GET request. Jika koneksi berhasil, data akan dikirim, dan respon dari server akan ditampilkan.

```
// Fungsi untuk mengirim data ke gateway
void sendNoiseData(float noise) {
  if (WiFi.status() == WL_CONNECTED) { // Pastikan koneksi Wi-Fi terhubung
    HTTPClient http; // Membuat objek HTTP untuk koneksi
    String url = "http://" + String(gatewayIP) + "/data?noise=" + String(noise); //
    http.begin(url); // Memulai koneksi HTTP ke URL yang telah dibentuk

 int httpResponseCode = http.GET(); // Mengirimkan data kebisingan dengar
 if (httpResponseCode > 0) {
   Serial.print("Data terkirim, kode respons: "); // Menampilkan respons
   Serial.println(httpResponseCode);
 } else {
   Serial.print("Error pengiriman data, kode: "); // Jika terjadi kesalah
   Serial.println(httpResponseCode);
 }
 http.end(); // Mengakhiri koneksi HTTP
} else {
 Serial.println("WiFi tidak terhubung."); // Menampilkan pesan jika Wi-Fi
```

Di bagian ini, kode memeriksa status koneksi Wi-Fi sebelum memulai pengiriman data. Jika terhubung, objek HTTPClient digunakan untuk membuat permintaan HTTP GET dengan URL yang telah dibentuk. Kode juga memeriksa kode respons dari server dan mencetak hasilnya di Serial Monitor. Jika koneksi tidak terhubung, kode akan memberikan pesan kesalahan.

4. Output Hasil pengiriman Data Perpustakaan

Setelah menulis kode untuk esp pengirim Perpustakaan, lakukan run pada Arduino IDE untuk melihat hasil print serial monitor seperti Gambar III.10.

Gambar III.10 Output Hail pengiriman Data Perpustakaan

e. Konfigurasi Kode ESP32 Gateway untuk Menjadi Mode Transmitter

ESP32 yang berfungsi sebagai gateway dalam mode Transmitter dirancang untuk menerima data dari perangkat sensor kebisingan yang terhubung (seperti ESP32 di masjid dan perpustakaan) dan meneruskan data tersebut ke server atau platform ThingSpeak untuk dapat di visualisasikan.

1. Koneksi Wi-Fi dan Inisialisasi

Di bagian awal kode, kita melakukan inisialisasi koneksi Wi-Fi dan mengatur server HTTP.

```
const char* ssid = "WIFI_RANIRY"; // SSID baru tanpa password
const char* password = ""; // Tanpa password

WebServer server(80); // Deklarasi objek server pada port 80

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password); // Memulai koneksi Wi-Fi

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println();
  Serial.print("Terhubung ke Wi-Fi! IP Address: ");
  Serial.println(WiFi.localIP());

  server.on("/data", HTTP_GET, handleData); // Menangani permintaan GET di e
  server.begin(); // Memulai server
}
```

Di bagian ini, kode mengatur koneksi Wi-Fi menggunakan SSID "WIFI_RANIRY". Setelah terhubung, server web diaktifkan pada port 80 untuk menerima permintaan HTTP. Endpoint /data ditentukan untuk menangani permintaan GET.

2. Menerima dan Memproses Data Kebisingan

Fungsi handleData() memproses permintaan data kebisingan dari klien. Kode ini memeriksa apakah ada argumen noise, mengubah format nilai, dan menyimpan data kebisingan berdasarkan alamat IP pengirim (Masjid atau Perpustakaan). Respon dikirim kembali kepada pengirim.

```
void handleData() {
  IPAddress clientIP = server.client().remoteIP(); // Mendapatkan alamat IP klien
  Serial.print("Alamat IP klien: ");
  Serial.println(clientIP); // Menampilkan alamat IP klien

  if (server.hasArg("noise")) {
    String noiseValue = server.arg("noise"); // Mengambil nilai dari argumen "noise"
    noiseValue.replace(",", "."); // Mengganti ',' dengan '.' jika ada
    float noise = noiseValue.toFloat(); // Mengonversi nilai ke float

  // Cek dari mana data datang dan simpan nilainya
  if (clientIP == IPAddress(192, 168, 201, 212)) { // Alamat IP pengirim Masjid
    noiseMosque = noise + 30.0; // Menambahkan 30 dB
    Serial.print("Data diterima dari Masjid: ");
    Serial.println(noiseMosque);
  } else if (clientIP == IPAddress(192, 168, 201, 153)) { // Alamat IP pengirim Perp
    noiseLibrary = noise + 30.0; // Menambahkan 30 dB
    Serial.print("Data diterima dari Perpustakaan: ");
    Serial.println(noiseLibrary);
  } else {
    Serial.println("Data diterima dari sumber yang tidak dikenal.");
  }

  // Mengirimkan respons kembali ke pengirim
  server.send(200, "text/plain", "Data diterima"); // Mengirim respons sukses
} else {
  server.send(400, "text/plain", "Parameter tidak valid"); // Mengirim respons error
}
```

3. Menampilkan dan Mengirim Data ke ThingSpeak

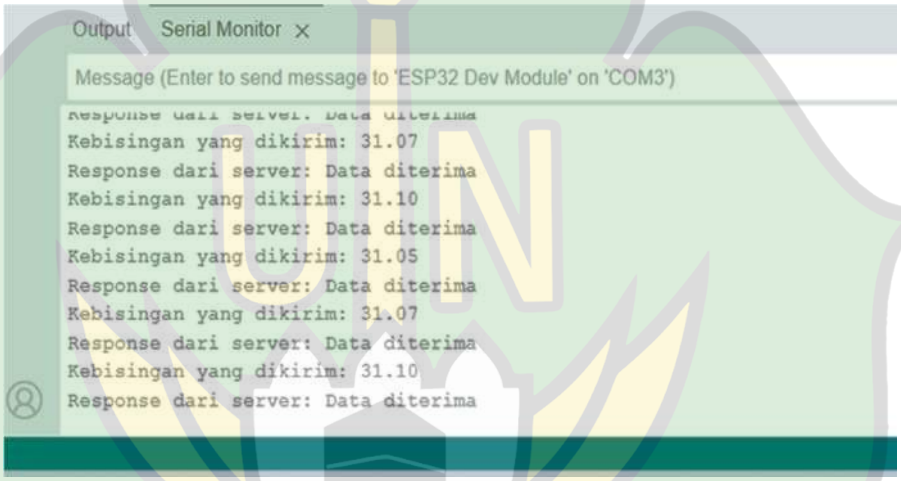
Bagian ini mengirimkan data yang dibaca ke server ThingSpeak dan menampilkan hasilnya:

```
void sendToThingSpeak(float noiseMosque, float noiseLibrary) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

    String url = String("http://") + thingSpeakHost + "/update?api_key=" + thingSpeakA
                 "&field1=" + String(noiseMosque) +
                 "&field2=" + String(noiseLibrary);

    http.begin(url); // Memulai koneksi HTTP
    int httpResponseCode = http.GET(); // Mengirim permintaan GET

    if (httpResponseCode > 0) {
      String response = http.getString(); // Mendapatkan respon dari ThingSpeak
      Serial.println(httpResponseCode); // Menampilkan kode respon
      Serial.println(response); // Menampilkan respon
    } else {
      Serial.print("Error on sending GET: ");
      Serial.println(httpResponseCode); // Menampilkan kode error
    }

    http.end(); // Menutup koneksi
  } else {
    Serial.println("WiFi not connected");
  }
}
```

Fungsi sendToThingSpeak() mengirimkan data kebisingan yang telah diterima ke platform ThingSpeak setiap 30 detik. Kode ini memeriksa status koneksi Wi-Fi, membangun URL untuk permintaan HTTP GET dengan API Key yang diberikan, dan mencetak respons dari ThingSpeak untuk debugging.

4. Hasil Output ESP32 sebagai Transmitter

Setelah menuliskan kode ESP sebagai Transmiter run kode agar mengeluarkan Outpuut di serial monitor seperti Gambar III.11.



Gambar III.11 Hasil Output ESP32 sebagai Transmitter

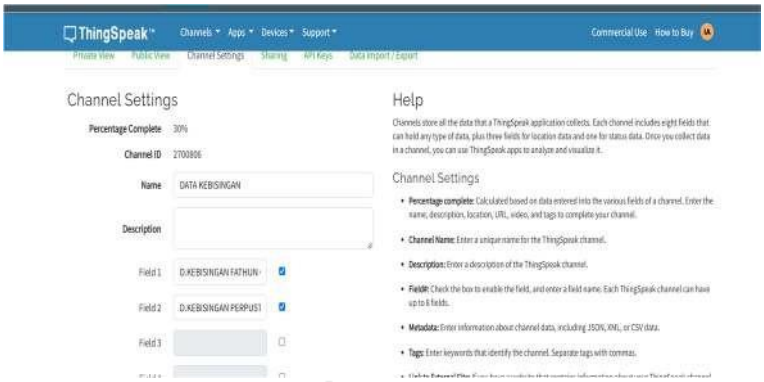
f. Visualisasi Data di ThingSpeak

1. Membuat Akun di ThingSpeak

Daftar untuk membuat akun jika belum memiliki satu. Anda akan menerima email konfirmasi.

2. Membuat Channel Baru

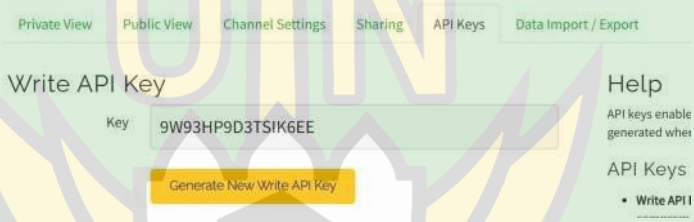
Untuk memulai, login ke akun Anda dan navigasikan ke dashboard. Di menu atas, klik pada opsi Channels, lalu pilih New Channel. Isi nama channel dan deskripsi, misalnya, Monitoring Kebisingan. Selanjutnya, buat dua field: Field 1 dengan nama "D.KEBISINGAN FATHUN QARIB" dan Field 2 dengan nama D.KEBISINGAN PERPUSTAKAAN. Setelah semua informasi terisi dengan benar, klik "Save Channel" untuk menyimpan pengaturan channel anda, seperti pada Gambar III.12 pengaturan channel Anda.



Gambar III.12 Pengaturan Channel Anda

3. Mendapatkan API Key

- Setelah membuat channel, Anda akan dibawa ke halaman detail channel.
- Catat API Key (akan digunakan dalam kode Anda). Berikut adalah tampilan API Key seperti Gambar III.13.



Gambar III.13 Tampilan API Key

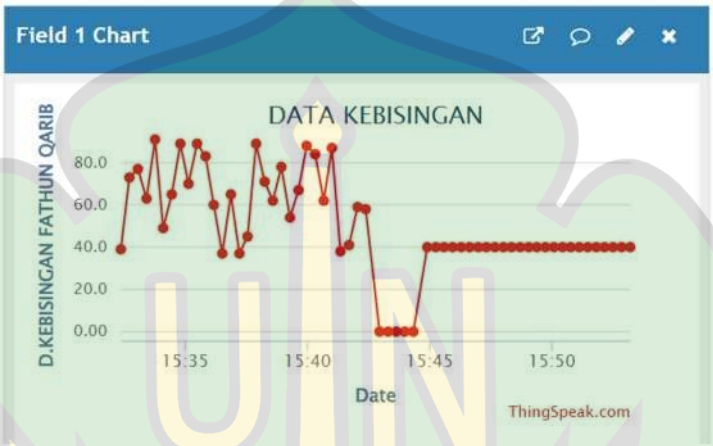
4. Membuat Visualisasi

- Klik "Add Widgets" untuk menambahkan visualisasi.
- Sesuaikan widget sesuai kebutuhan Anda dan klik "Save".seperti Gambar III.14.

Gambar III.14 Widget

5. Melihat Data secara Real-Time

- Setelah data mulai dikirim dari ESP32 Anda, Anda akan melihat visualisasi di halaman channel.
- Refresh halaman untuk melihat data tampilan terbaru.seperti pada Gambar III.15.



Gambar III.15 Tampilan Dasboar

BAB IV
HASIL DAN PEMBAHASAN

IV.1 Implementasi Sistem Half Duplex

Dalam penelitian ini, sistem pemantauan kebisingan berbasis IoT dirancang untuk berfungsi sebagai komunikasi Half-Duplex. Proses dimulai dengan pengumpulan data dari sensor kebisingan yang terpasang di dua lokasi, yaitu Masjid dan Perpustakaan UIN Ar-Raniry, Setiap sensor terhubung dengan ESP32 yang bertindak sebagai pengirim data. Sensor mengukur tingkat kebisingan di lingkungannya dan mengirimkan informasi tersebut ke ESP32 Gateway melalui protokol HTTP. Pada tahap ini, komunikasi bersifat satu arah, di mana satu pengirim dapat mengirim data ke ESP32 Gateway pada satu waktu.

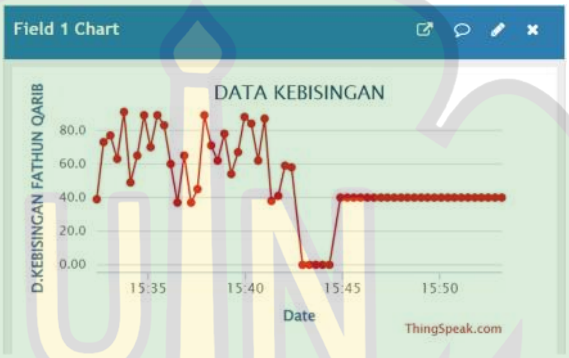
Setelah ESP32 Gateway menerima data dari kedua lokasi, ia mengolah dan mengirimkan data tersebut ke ThingSpeak untuk penyimpanan dan analisis lebih lanjut. Proses pengiriman data dilakukan setelah menunggu data dari pengirim diterima, sehingga memastikan bahwa komunikasi berlangsung secara bergantian dan tidak simultan. Dengan demikian, sistem ini menghindari tabrakan data antar pengirim. Pengaturan ini menegaskan metode komunikasi Half-Duplex, di mana hanya satu jalur komunikasi yang aktif pada satu waktu, memungkinkan pemantauan kebisingan secara efisien dan real-time.

IV.2 Pengujian Kinerja Sistem

Pengujian kinerja sistem dilakukan untuk mengevaluasi keandalan dan efisiensi pengiriman data dari masing-masing ESP Pengirim ke ESP Gateway. Setiap pengirim mengirimkan data secara berkala. Data tingkat kebisingan yang terdeteksi dikirim ke ESP Gateway untuk diproses lebih lanjut.

IV.2.1 Grafik Hasil Tingkat Kebisingan Di Mesjid Fathun Qarib

Grafik hasil tingkat kebisingan di Mesjid Fathun Qarib mencerminkan data yang diperoleh dari sistem pemantauan kebisingan berbasis IoT yang telah diterapkan dalam penelitian ini. Pengukuran dilakukan secara real-time menggunakan sensor kebisingan yang terhubung dengan ESP32 Gateway, yang mengumpulkan dan mengirimkan data ke platform visualisasi.Hasil pemantauan kebisingan di Mesjid Fathun Qarib dapat di lihat pada Gambar IV.1.



Gambar IV.1 Kebisingan di Mesjid Fathun Qarib

Grafik di atas menunjukkan bahwa tingkat kebisingan suara di masjid relevan stabil, tetapi pada saat memasuki waktu sholat Ashar, sensor mendeteksi suara yang cenderung lebih besar, di antara 40 dB ke atas, sampai jam 15.43, dan kembali stabil di angka 40 dB ketika jam menunjukkan pukul 15.45 sampai jam 15.50 lewat.

IV.2.2 Grafik Hasil Tingkat Kebisingan Di Perpustakaan

Grafik hasil tingkat kebisingan di Perpustakaan UIN AR-RANIRY memberikan gambaran yang jelas tentang dinamika kebisingan dalam lingkungan belajar yang vital ini. Data yang disajikan diambil dari sensor kebisingan yang terintegrasi dengan sistem pemantauan berbasis IoT, yang mengandalkan ESP32 Gateway untuk mengumpulkan dan mengirimkan informasi secara real-time. Hasil pemantauan kebisingan di perpustakaan UIN Ar-raniry dapat di lihat pada Gambar IV.2.

.

Gambar IV.2 .kebisingan di Perpustakaan

Grafik di atas menunjukkan bahwa tingkat kebisingan suara perpustakaan relatif stabil. Di angka 31 dB.ini menunjukan tidak ada perubhan suara yang sigifikan dalm uangan perpustakaan.

IV.3 Hasil Pembacaan Rata-Rata Sensor

Setelah melakukan pengetesan alat, dikarenakan akun platform gratis yang membatasi hanya boleh membuat 4 kanal, maka disimpulkan bagaimana sistem ini dapat bekerja sesuai rencana. Beberapa nilai rata-rata kebisingan yang dikumpulkan oleh sensor di ESP pengirim 1 di Masjid Fathun Qarib dan ESP pengirim 2 di Perpustakaan berhasil dikumpulkan menggunakan sistem Half-Duplex dengan menggunakan ESP32. Hasil nilai rata-rata kebisingan dapat dilihat pada Tabel IV.1.

Tabel IV.1 Hasil Rata-Rata Nilai Kebisingan

WAKTU	TEMPAT		SISTEM HALF-DUPLEX (SUKSES/GAGAL)
	MESJID FATHUN QARIB(dB)	PERPUSTAKAN (dB)	
2024-10-27 08:12:37	30	31	Sukses
2024-10-27 08:12:58	30	31	Sukses
2024-10-27 08:13:18	30	31	Sukses
2024-10-27 08:13:39	30	31	Sukses
2024-10-27 08:14:00	30	31	Sukses
2024-10-27 08:14:20	30	31	Sukses
2024-10-27 08:14:39	60	61	Sukses
2024-10-27 08:15:00	60	61	Sukses

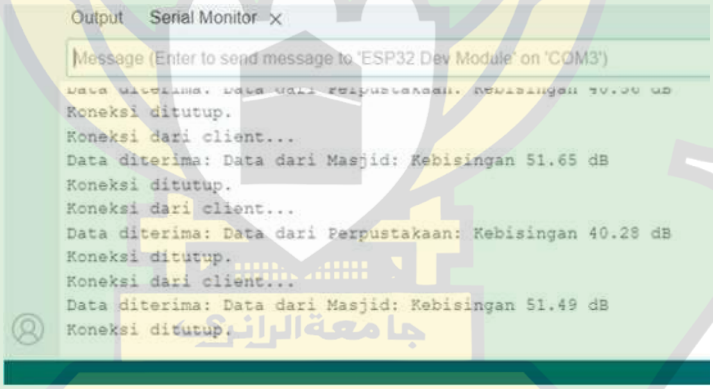
2024-10-27 08:15:17	60	61	Sukses
2024-10-27 08:15:34	60	61	Sukses
2024-10-27 08:15:52	60	61	Sukses
2024-10-27 08:16:08	60	61	Sukses
2024-10-27 08:16:25	60	61	Sukses
2024-10-27 08:16:43	60	61	Sukses
2024-10-27 08:16:59	60	61	Sukses
2024-10-16 11:01:04	30	31	Sukses
2024-10-16 11:01:02	30	31	Sukses
2024-10-16 11:01:04	30	31	Sukses
2024-10-16 11:01:06	30	31	Sukses
2024-10-16 11:01:27	30	31	Sukses
2024-10-16 11:01:47	30	31	Sukses
2024-10-16 11:02:08	30	31	Sukses
2024-10-16 11:02:29	30	31	Sukses
2024-10-16 11:02:50	30	31	Sukses
2024-10-16 11:03:10	30	31	Sukses
2024-10-16 11:03:31	30	31	Sukses
2024-10-16 11:03:52	30	31	Sukses
2024-10-16 11:04:12	30	31	Sukses
2024-10-16 11:04:33	30	31	Sukses
2024-10-16 11:00:04	30	31	Sukses
2024-10-16 11:00:25	30	31	Sukses
2024-10-16 11:00:46	30	31	Sukses
2024-10-27T15:44:54	40	50	Sukses
2024-10-27T15:45:15	40	50	Sukses
2024-10-27T15:45:35	40	50	Sukses
2024-10-27T15:45:56	40	50	Sukses
2024-10-27T15:46:17	40	50	Sukses
2024-10-27T15:46:38	40	50	Sukses
2024-10-27T15:46:59	40	50	Sukses
2024-10-27T15:47:19	40	50	Sukses
2024-10-27T15:47:40	40	50	Sukses
2024-10-27T15:48:01	40	50	Sukses
2024-10-27T15:48:21	40	50	Sukses
2024-10-27T15:48:42	40	50	Sukses
2024-10-27T15:49:03	40	50	Sukses
2024-10-27T15:49:23	40	50	Sukses
2024-10-27T15:49:44	40	50	Sukses
NILAI RATA RATA	**PERPUSTAKAAN**	**MASJID**	**STANDAR KEMENTERIAN**
	38,64 dB	**40,91 dB**	55 dB (Perpustakaan) dan 55 dB(Tempat ibadah)

IV.4 Analisis Hasil Kinerja Half-Duplex

Hasil pengujian sistem Half-Duplex menunjukkan bahwa kedua ESP Pengirim, yaitu yang terletak di Masjid Fathun Qarib dan Perpustakaan UIN AR-RANIR, berhasil mengirimkan data kebisingan dengan efisiensi yang tinggi. Latensi pengiriman data yang rendah menjadi salah satu indikator utama keberhasilan sistem ini, yang diukur berdasarkan waktu yang diperlukan untuk mengirimkan data dari masing-masing ESP Pengirim ke ESP Gateway.

a. Pengaruh Jumlah perangkat yang terhubung

Penelitian ini melihat pengaruh jumlah perangkat dalam sistem half duplex menunjukkan bahwa peningkatan jumlah ESP Pengirim dapat berpengaruh pada latensi dan keandalan pengiriman data. Dalam pengujian ini, hanya dua perangkat yang digunakan, yaitu ESP Pengirim di Masjid Fathun Qarib dan di Perpustakaan UIN AR-RANIR. Kondisi ini terbukti optimal untuk sistem half duplex yang diimplementasikan, di mana masing-masing perangkat dapat berfungsi dengan baik dalam mengirimkan data.Seperti pada Gambar Output Serial Monitor IV.3.

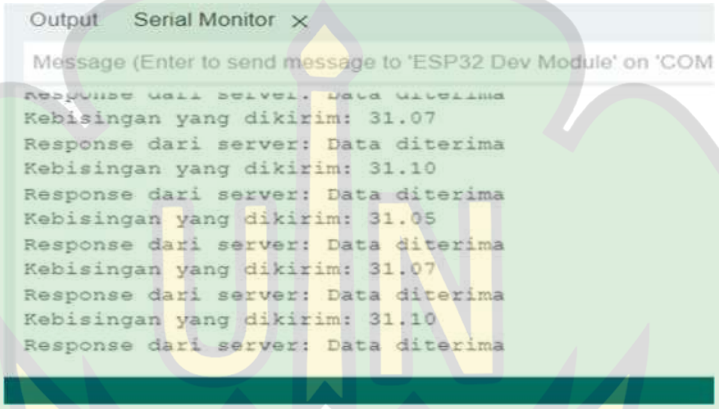


Gambar Output Serial Monitor IV.3

Namun, penambahan lebih banyak perangkat berpotensi menyebabkan penurunan dalam keandalan pengiriman dan peningkatan latensi. Hal ini disebabkan oleh karakteristik sistem Half-Duplex yang mengharuskan perangkat untuk bergantian dalam proses pengiriman dan penerimaan data. Ketika lebih banyak perangkat berusaha untuk berkomunikasi secara bersamaan, konflik dalam pengiriman data dapat terjadi, yang berpotensi mengakibatkan kehilangan data dan keterlambatan dalam pengiriman.

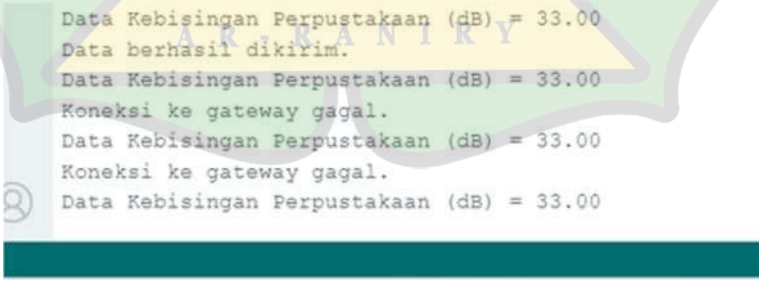
b. Kondisi jaringan Wi-Fi

Kondisi jaringan Wi-Fi sangat berpengaruh di mana sistem beroperasi juga memainkan peranan penting dalam kinerja sistem. Pengujian yang dilakukan dalam kondisi jaringan yang stabil menunjukkan bahwa data berhasil di terima dan dikirim dengan cepat dan tanpa gangguan. Keberhasilan ini mencerminkan efektivitas sistem dalam kondisi optimal. Seperti pada Gambar IV.5.



Gambar IV.5 Output Kondisi Jaringan Stabil

Namun, dalam situasi di mana sinyal jaringan lemah atau terdapat interferensi, hasil pengujian menunjukkan bahwa latensi dapat meningkat, dan keandalan pengiriman data dapat menurun.Seperti pada Gambar IV.6.



Gambar IV.6 Output Kondisi Jaringan Buruk

VI.5 Kendala dan Tantangan Penelitian

Selama pelaksanaan penelitian ini, terdapat beberapa kendala dan tantangan yang dihadapi, antara lain:

1. Keterbatasan Sumber Daya Perangkat

   Jumlah perangkat yang tersedia untuk percobaan ini sedikit menjadi salah satu kendala utama. Beberapa unit ESP32 yang digunakan dalam penelitian ini bukanlah perangkat standar industri, melainkan model untuk pelajar atau pemula. Oleh karena itu, ESP32 cenderung mengalami panas berlebih jika dipakai dalam waktu lama yang mengakibatkan kinerjanya menurun, sehingga proses uji coba dan pengujian sistem menjadi lebih sulit berjalan lancar. Untuk mendapatkan hasil terbaik, proses ini memerlukan waktu yang lebih lama.

2. Kendala Jaringan dan Konektivitas

   Mengingat penelitian ini melibatkan pemantauan data secara real-time melalui jaringan Wi-Fi, kendala terkait kestabilan dan kekuatan sinyal Wi-Fi di lokasi penelitian, yaitu Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry, sempat menghambat pengiriman data secara kontinu. Beberapa kali terjadi gangguan koneksi yang mempengaruhi keakuratan pengukuran data.

3. Kompleksitas Integrasi Sistem

   Integrasi antara perangkat keras (ESP32, sensor, dan jaringan) dengan platform ThingSpeak memerlukan pemrograman yang kompleks. Pengaturan komunikasi half-duplex dan pengolahan data secara real-time juga menjadi tantangan dalam memastikan data dapat diproses dan ditampilkan dengan akurat tanpa adanya keterlambatan.

4. Waktu dan Sumber Daya Manusia

   Keterbatasan waktu operasional Perpustakaan dan Masjid serta sumber daya manusia dalam melakukan pemantauan dan pengujian secara intensif di dua lokasi yang berbeda juga menjadi tantangan.

BAB V
KESIMPULAN DAN SARAN

V.1 Kesimpulan

Dari hasil Penelitian "Perancangan Sistem Pemantauan Kebisingan berbasis IoT dengan metode komunikasi Half-Duplex menggunakan ESP32 di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry", maka didapatkan kesimpulan sebagai berikut:

1. Penelitian ini berhasil merancang dan mengimplementasikan sistem pemantauan kebisingan berbasis IoT yang memanfaatkan komunikasi Half-Duplex, dengan menggunakan ESP32 sebagai gateway yang menghubungkan sensor kebisingan di Masjid Fathun Qarib dan Perpustakaan UIN Ar-Raniry.

2. ESP32 terbukti efektif berfungsi sebagai gateway yang mengelola koneksi kedua sensor melalui komunikasi Half-Duplex. Namun, performa transfer data dipengaruhi oleh keterbatasan jaringan Wi-Fi kampus, yang berpotensi memperlambat pengiriman data pada jam-jam sibuk.

3. ESP32 berhasil berfungsi ganda sebagai penerima dan pengirim data dari sensor, serta mengirimkan data tersebut ke platform ThingSpeak untuk divisualisasikan secara real-time, sesuai dengan tujuan penelitian.

4. Berdasarkan data yang diperoleh di tabel penelitian diatas , tingkat kebisingan di perpustakaan sebesar 38,64 dB dan di masjid sebesar 40,91 dB. dibawah Standar kebisingan yang ditetapkan oleh kementerian yaitu 55 dB untuk perpustakaan dan tempat ibadah.

Secara keseluruhan, sistem ini terbukti efektif untuk pemantauan kebisingan di lingkungan kampus, memberikan solusi yang efisien dan terjangkau.

V.2 Saran

Hasil penelitian ini lebih fokus pada teknis bagaimana sistem Half-Duplex ini bisa dijalankan menggunakan ESP32, yang dapat berfungsi sebagai penerima dan pengirim data untuk divisualisasikan. Penggunaan perangkat yang terbatas dan koneksi Wi-Fi kampus yang terkadang buruk mungkin menjadi masalah tersendiri bagi penulis selama melakukan penelitian dan pengujian sistem ini. Penulis berharap penelitian ini dapat dikembangkan lebih baik ke depannya, dengan menggunakan perangkat yang lebih baik, seperti ESP yang lebih canggih dan perangkat lainnya, sehingga diharapkan dapat menghasilkan sistem yang juga lebih baik.

DAFTAR PUSTAKA

Abilovani, Zavero Brillianata, Widhi Yahya, dan Fariz Andri Bakhtiar. 2018. *Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT*. Vol. 2.

Andianingsari, Diah, dan Abdul Rahman. 2023. *PEMANTAUAN KEBISINGAN DENGAN MENGGUNAKAN PETA KEBISINGAN PROGRAM SUFFER 15 DI PT. F*. Vol. 04.

Theodorus+S+Kalengkongan. ". "pengaruhkebisinganke lingkungan."

Kolaborasi, Jurnal, dan Resolusi Konflik. t.t. "PENGARUH KEPADATAN PENDUDUK TERHADAP TINDAKAN KRIMINAL." 3.

Lukita Sari, Ardila, dan Eksa Lailia Maulidina. 2022. *ALAT PENDETEKSI KEBOCORAN GAS LPG DENGAN MEMANFAATKAN TELEGRAM BOT SEBAGAI MEDIA INFORMASI LPG GAS LEAKAGE DETECTOR BY UTILIZING BOT TELEGRAM AS INFORMATION*. Vol. 15.

Muhamad Ilham Faozi, dan Thufail Dhiva Arga Nugraha. 2022. "Teknologi dan Destruktivitas Manusia." *Jurnal Multidisiplin Madani* 2(5):2079–94. doi: 10.55927/mudima.v2i5.257.

Mujayin Kholik Dan Dimas Adji Krishna, Heri. t.t. *ANALISIS TINGKAT KEBISINGAN PERALATAN PRODUKSI TERHADAP KINERJA KARYAWAN*.

Nabawi, I'sham Sajid, dan Rohjai Badarudin. 2024. "PERANCANGAN SISTEM KONTROL LAMPU BERBASIS MIKROKONTROLER ARDUINO UNO R3 DENGAN SENSOR SUARA LM393, ELECTRET MICROPHONE." *Jurnal*

Informatika dan Teknik Elektro Terapan 12(3). doi: 10.23960/jitet.v12i3.4536. Nahdi, Fahad, dan Harry Dhika. t.t. *Analisis Dampak Internet of Things (IoT) Pada Perkembangan Teknologi di Masa Yang Akan Datang 33*.

Sari, Indah Purnama, Aisar Novita, Al-Khowarizmi Al-Khowarizmi, Fanny Ramadhani, dan Andy Satria. 2024. "Pemanfaatan Internet of Things (IoT) pada Bidang Pertanian Menggunakan Arduino UnoR3." *Blend Sains Jurnal Teknik* 2(4):337–43. doi: 10.56211/blendsains.v2i4.505.

Tri Putra, Agung, dan Jl Hamka Air Tawar. 2021. *Penggunaan Aplikasi Ubidots untuk Sistem Kontrol dan Monitoring pada Gudang Gula Berbasis Arduino UNO*. Vol. 2.

Wilani, Lia, Mardian Peslinof, Jesi Pebralia, Program Studi Fisika, Fakultas Sains dan Teknologi, dan Universitas Jambi. t.t. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi) RANCANG BANGUN SISTEM*

Arifin, M. N., Hannats, M., Ichsan, H., & Akbar, S. R. (2018). *Monitoring Kadar Gas Berbahaya Pada Kandang Ayam Dengan Menggunakan Protokol HTTP Dan ESP8266* (Vol. 2, Issue 11). http://j-ptiik.ub.ac.id

Mahanin Tyas, U., Apri Buckhari, A., Studi Pendidikan Teknologi Informasi, P., & Studi Pendidikan Teknologi dan Kejuruan, P. (2023). *IMPLEMENTASI APLIKASI ARDUINO IDE PADA MATA KULIAH SISTEM DIGITAL* (Vol. 1, Issue 1).

Ningsih, N., Ramadhani, A. D., Nurcahya, A., & Azizah, N. (2022). Klasifikasi dan Monitoring Kualitas Udara Dalam Ruangan menggunakan Thingspeak. In *Afifah Dwi Ramadhani* (Vol. 10, Issue 1). https://journal.trunojoyo.ac.id/triac

Prayudha, J., Pranata, A., & Prastyo, H. (2020). J-SISKO TECH Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Implementasi Teknik Komunikasi Serial Half Duplex Pada Kendali Jarak Jauh Lampu Ruangan Rumah Berbasis Internet Of Things (IOT). , *32*(1), 32–40.

Satriawan, I., Haryanto, T., & Wijaya, R. (2020). Pemantauan Kebisingan Berbasis IoT untuk Pengelolaan Lingkungan Kota. *Jurnal Teknologi dan Sistem Informasi*, 15(2), 99-107.

Nugroho, D., & Pratama, R. (2018). Optimasi Komunikasi Data dengan Metode Half-Duplex dalam Sistem IoT. *Jurnal Ilmiah Teknologi Komunikasi*, 9(3), 45-

.LAMPIRAN

- Lampiran 1 Hasil Tracerouter

  Tabel ini adlah data untuk mengetahui jalur jaringan dari tiga ESP32: ESP pengirim 1 di Masjid Fathun Qarib, Perpustakaan, dan ESP Gateway.

TRACERT ESP MESJID THINGSPEAK	TRACERT ESP PERPUS-THINGSPEAK	ESP GATEWAY TO THINGSPEAK
1 35 ms 56 ms 23 ms 192.168.104.1	1 15 ms 4 ms 5 ms 192.168.201.1	1 607 ms 200 ms 179 ms 192.168.201.1
2 14 ms 7 ms 59 ms 10.20.0.1	2 3 ms 3 ms 5 ms 10.20.0.5	2 700 ms 50 ms 148 ms 10.20.0.5
3 50 ms 9 ms 34 ms 103.107.187.1	3 3 ms 3 ms 7 ms 103.107.187.1	3 595 ms 382 ms 292 ms 103.107.187.1
4 3 ms 3 ms 8 ms 192.168.116.2	4 6 ms 3 ms 12 ms 192.168.116.2	4 17 ms 35 ms 27 ms 192.168.116.2
5 * 23 ms 19 ms 36.64.254.197	5 18 ms 18 ms 15 ms 36.64.254.197	5 114 ms 23 ms 83 ms 36.64.254.197
6 * * * Request timed out.	6 32 ms * 21 ms 180.240.193.93	6 245 ms 26 ms 240 ms 180.240.193.93
7 * * * Request timed out.	7 * 20 ms * 180.240.193.93	7 * 773 ms * 180.240.193.93
8 249 ms 249 ms 255 ms 180.240.192.89	8 368 ms 335 ms 266 ms 180.240.192.89	8 466 ms 526 ms 271 ms 180.240.192.89
9 253 ms 252 ms 250 ms equinix02-iad2.amazon.com [206.126.236.35]	9 261 ms 276 ms 274 ms equinix02-iad2.amazon.com [206.126.236.35]	9 3661 ms 2367 ms 1722 Ms equinix02iad2.amazon.com [206.126.236.35]
10 * * * Request timed out.	10 * * * Request timed out.	10 * * * Request timed out.
11 * * * Request timed out.	11 * * * Request timed out.	11 * * * Request timed out.
12 * * * Request timed out.	12 * * * Request timed out.	12 * * * Request timed out.
13 * * * Request timed out.	13 * * * Request timed out.	13 * * * Request timed out.
14 * * * Request timed out.	14 * * * Request timed out.	14 * * * Request timed out.
15 * * * Request timed out.	15 * * * Request timed out.	15 * * * Request timed out.
16 * * * Request timed out.	16 * * * Request timed out.	16 * * * Request timed out.
17 * * * Request timed out.	17 * * * Request timed out.	17 * * * Request timed out.
18 * * * Request timed out.	18 * * * Request timed out.	18 * * * Request timed out.
19 * * * Request timed out.	19 * * * Request timed out.	19 * * * Request timed out.
20 * * * Request timed out.	20 * * * Request timed out.	20 * * * Request timed out.
21 * * * Request timed out.	21 * * * Request timed out.	21 * * * Request timed out.
22 * * * Request timed out.	22 * * * Request timed out.	22 * * * Request timed out.
23 * * * Request timed out.	23 * * * Request timed out.	23 * * * Request timed out.
24 * * * Request timed out.	24 * * * Request timed out.	24 * * * Request timed out.
25 * * * Request timed out.	25 * * * Request timed out.	25 * * * Request timed out.
26 * * * Request timed out.	26 * * * Request timed out.	26 * * * Request timed out.
27 * * * Request timed out.	27 * * * Request timed out.	27 * * * Request timed out.
28 * * * Request timed out.	28 * * * Request timed out.	28 * * * Request timed out.
29 * * * Request timed out.	29 * * * Request timed out.	29 * * * Request timed out.
30 * * * Request timed out.	30 * * * Request timed out.	30 * * * Request timed out.

- Lampiran 2 Foto Pelaksanaan Penelitian

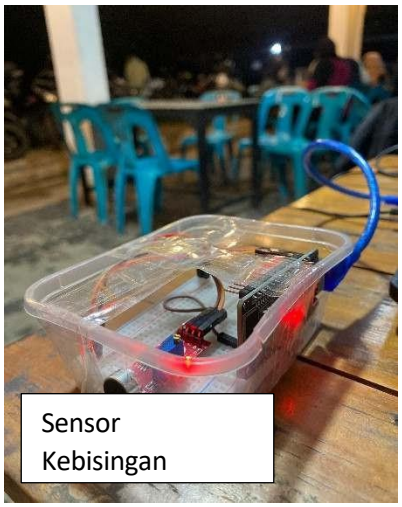
ESP Pengirim 1 Mesjid

Setting ESP Pengirim 1 Mesjid

Sensor Kebisingan di Mesjid

ESP Pengirim 2 Pepustakaan

Hasil Serial Monitor ESP



Sensor Kebisingan