

**ALAT PENDETEKSI KEBISINGAN DAN PEMBERI PERINGATAN DI
PERPUSTAKAAN BERBASIS WEB SERVER**



SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar
Sarjana Teknik pada Program Studi S1 Teknik Elektro Jurusan Teknik Elektro
Fakultas Teknik Universitas Tadulako

Disusun oleh:

Dicky Christian Saul
F 441 20 063

PROGRAM STUDI S1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS TADULAKO
PALU
2025

**NOISE DETECTION AND ALERT SYSTEM IN A WEB
SERVER-BASED LIBRARY**



THESIS

Submitted as a partial fulfillment of the requirements for Bachelor Degree
at Electrical Engineering Department
Faculty of Engineering
Tadulako University

Disusun oleh:

Dicky Christian Saul
F 441 20 063

**ELECTRICAL ENGINEERING DEPARTMENT
ENGINEERING FACULTY
TADULAKO UNIVERSITY
PALU
2025**



HALAMAN PENGESAHAN

Berdasarkan persetujuan dari Panitia Ujian Skripsi Program Studi S1 Teknik Elektro, Jurusan Teknik Elektro Fakultas Teknik Universitas Tadulako, maka judul:

"ALAT PENDETEKSI KEBISINGAN DAN PEMBERI PERINGATAN DI PERPUSTAKAAN BERBASIS *WEB SERVER*"



Disusun oleh:

DICKY CHRISTIAN SAUL
NIM. F 441 20 063



Disahkan oleh:

Dekan Fakultas Teknik
Universitas Tadulako

Ketua Jurusan
Teknik Elektro



Ir. Andi Arham Adam, S.T., M.Sc(Eng), Ph.D
NIP. 19740323 199903 1 002



Dr. Ahamad Antares Adam, S.T., M.Eng.Sc
NIP. 19680420 199412 1 001

HALAMAN PERSETUJUAN

Pada hari Jumat, tanggal Dua Puluh Dua, bulan September, tahun Dua Ribu Dua Puluh Empat, Panitia Ujian Skripsi Program Studi S1 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako, berdasarkan SK No. 15008/UN.28.1.31/DK/2024, tanggal Enam, bulan Juni, tahun Dua Ribu Dua Puluh Tiga. Menyatakan menerima, menyetujui Skripsi yang telah dipertanggung jawabkan dihadapan Panitia Ujian Skripsi oleh :

DICKY CHRISTIAN SAUL
NIM. F 441 20 063

Judul :

**ALAT PENDETEKSI KEBISINGAN DAN PEMBERI PERINGATAN DI
PERPUSTAKAAN BERBASIS *WEB SERVER***

Menyetujui :

PEMBIMBING UTAMA



Ir. Tan Suryani Solli, MT
NIP.1965121799220001

PEMBIMBING PENDAMPING



Muh. Aristo Indrajaya S.T., M.T
NIP.199102102023211025

Palu, 1 Desember 2025
Ketua Program Studi S1 Teknik Elektro



Dr. Ir. Alamsyah, ST., MT
NIP.1974041420021210001

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : **Dicky Christian Saul**

No. Stambuk : **F 441 20 063**

Judul Skripsi : **Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server**

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah skripsi ini dan disebutkan dalam daftar pustaka.

Palu, 23 November 2025

Penulis



Dicky Christian Saul

NIM. F 441 20 063

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yesus Kristus atas Kasih dan karunia-Nya sehingga skripsi dengan judul Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server dapat penulis susun dan selesaikan dengan baik.

Penulisan skripsi ini dimaksudkan untuk memenuhi syarat memperoleh gelar Sarjana Teknik pada Program Studi S1 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako.

Keberhasilan skripsi ini dapat diselesaikan berkat bantuan dan dorongan dari berbagai pihak yang telah memberikan referensi, gagasan, bimbingan dan berbagai dukungan lainnya. Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Andi Arham Adam, S.T., M.Sc., Ph.D. sebagai Dekan Fakultas Teknik, Universitas Tadulako.
2. Ibu Dr. Yuli Asmi Rahman, S.T., M.Eng. sebagai Wakil Dekan Bidang Akademik Fakultas Teknik, Universitas Tadulako.
3. Bapak Dr. Ahmad Antares Adam, S.T., M.Eng.Sc., sebagai Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako, dan sekaligus sebagai dosen wali yang telah membimbing dan memberikan saran-saran bermanfaat kepada penulis selama menjalani perkuliahan
4. Bapak Dr. Ir. Alamsyah, S.T., M.T., sebagai Ketua Program Studi S1 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako.
5. Ibu Ir. Mery Subito, M.T., sebagai Ketua KDK Teknik Elektronika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako.

6. Kepada Bapak Alm.Ir. Ardi Amir, ST., M.T. sebagai Dosen Wali yang telah membimbing dan memberikan saran-saran bermanfaat kepada penulis selama menjalani perkuliahan.
7. Kepada Ibu Ir. Tan Suryani Solli, M.T., selaku Dosen Pembimbing I yang telah memberikan arahan serta masukan yang bermanfaat bagi penulis.
8. Kepada Bapak Muh. Aristo indrajaya, S.T., M.T., selaku Dosen Pembimbing II yang telah memberikan arahan serta masukan yang bermanfaat bagi penulis.
9. Kepada Bapak Dr. Ir. Alamsyah, S.T., M.T., Bapak Ir. Rizana Fauzi, S.T., M.T., dan Bapak Aidynal Mustari, S.T., M.T., selaku Dosen Penguji yang senantiasa memberikan kritik dan saran yang baik kepada penulis.
10. Seluruh dosen pengajar di Jurusan Teknik Elektro yang telah memberikan ilmu dan wawasan berharga kepada penulis selama menempuh pendidikan.
11. Seluruh staf, laboran, dan teknisi di laboratorium Jurusan Teknik Elektro yang telah berkontribusi serta memberikan dukungan dalam proses penyusunan skripsi ini.
12. Seluruh staf akademik dan administrasi Jurusan Teknik Elektro yang telah memberikan semangat serta bantuan yang berarti bagi penulis.
13. Kepada orang tua, Ibu Sringah Sutardji, yang senantiasa memberikan dukungan serta doa tanpa henti, menjadi sumber motivasi dan semangat bagi penulis dalam menyelesaikan skripsi ini.
14. Keluarga besar penulis yang selama ini memberikan dukungan dalam banyak hal selama masa pendidikan penulis.

15. Kepada Gracella Angelina The yang senantiasa memberikan dukungan dan bantuan kepada penulis selama proses penelitian.
16. Kepada kawan karib Yudistiran Mansoba, Zaenal Arifin Warnoto, ST, Rahmat Hidayat Hamimu, Suriadi, dan Rnevans Togar Gultom yang telah memberikan banyak bantuan
17. Kepada anggota kos glen Keny Glen Fulan Balok, Rohan Lexy Passorong, Kevin sumbung, Adam Anugrah, Enos, Dan Arfahuz Zaimil Maufur yang telah memberikan banyak bantuan dan dorongan kepada penulis selama ini.
18. Arester Angkatan 2020, dan teman-teman seperjuangan lainnya yang tidak sempat penulis sebutkan satu persatu yang telah memberikan bantuan kepada penulis selama ini.

Penulis menyadari sepenuhnya bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, saya sangat mengharapkan masukan dan kritik yang membangun dari berbagai pihak. Penulis juga berharap skripsi ini dapat bermanfaat, baik bagi mahasiswa maupun masyarakat umum, sebagai bahan pembelajaran dan pengembangan ilmu pengetahuan.

Palu, 27 Oktober 2025
Penulis

Dicky Christian Saul
NIM. F 441 20 063

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN.....	v
KATA PENGANTAR	vi
DAFTAR ISI	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR RUMUS	xiii
DAFTAR SINGKATAN	xiv
DAFTAR SIMBOL DIAGRAM ALIR	xv
ABSTRAK	xvi
<i>ABSTRACT</i>	<i>xvii</i>
 BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Sistematika Penulisan	4
 BAB II TINJAUAN PUSTAKA	
2.1 Tinjauan Pustaka.....	5
2.2 Landasan Teori.....	9
2.2.1 Kebisingan	9
2.2.2 Perpustakaan	10
2.2.3 Sensor MAX9814.....	13
2.2.4 ESP32	14
2.2.5 <i>DFPlayer</i>	16
2.2.6 <i>Liquid Crystal Display</i>	17
2.2.7 <i>Light Emitting Diode</i>	18
2.2.8 Web Server.....	19
2.2.9 <i>Visual Studio Code</i>	23
2.2.10 <i>Arduino IDE</i>	23
2.2.11 <i>Desibel Suara</i>	23
 BAB III METODE PENELITIAN	
3.1 Alat dan Bahan Penelitian.....	25
3.1.1 Alat	25
3.1.2 Bahan	25
3.2 Tahapan Penelitian.....	26

3.2.1	Diagram Alir	26
3.2.2	Diagram Blok.....	29
3.2.3	Perancangan Sistem	29
BAB IV HASIL DAN PEMBAHASAN		
4.1	Bentuk Fisik Alat	35
4.2	Pengujian Sistem.....	36
4.2.1	Pengujian <i>LCD</i>	37
4.2.2	Pengujian Sensor Suara	40
4.2.3	Pengujian Konsumsi Daya	41
4.2.4	Pengujian Tampilan Web Dan <i>Real Time</i> Grafik.....	43
4.2.5	Pengujian Jangkauan Sensor	45
4.2.6	Pengujian Alat Keseluruhan	45
BAB V PENUTUP		
5.1	Kesimpulan	47
5.2	Saran.....	47
DAFTAR PUSTAKA		49
LAMPIRAN		51

DAFTAR GAMBAR

No. Gambar	Halaman
2.1 Perpustakaan	13
2.2 Sensor MAX9814	14
2.3 ESP32.....	15
2.4 <i>DFPlayer</i>	16
2.5 <i>Liquid Crystal Display</i>	18
2.6 <i>light Emitting Diode</i>	19
2.7 Web Server.....	20
2.8 <i>Arduino IDE</i>	23
3.1 Diagram Alir Penelitian	26
3.2 Diagram Blok Perancangan Sistem	28
3.3 Skema Rancangan Sistem	29
3.4 Diagram Alir Prinsip Kerja Alat	30
3.5 Halaman Beranda	32
3.6 Halaman Grafik Suara	32
3.7 Halaman <i>History</i> Suara	33
3.8 Rancangan Desain Tampak Depan.....	33
3.9 Rancangan Desainin Tampak Samping Kanan	34
4.1 Bentuk Fisik Alat.....	35
4.2 Diagram Pengujian Alat.....	37
4.3 <i>LCD</i> Saat Dalam Kondisi Tenang	38
4.4 <i>LCD</i> Saat Dalam Kondisi Berisik.....	39
4.5 <i>LCD</i> Saat Dalam Kondisi Sangat Berisik	40
4.6 Menu Grafik Suara	44
4.7 Tampilan Menu <i>History</i> Suara	44

DAFTAR TABEL

No. Tabel	Halaman
2.1 Perbandingan Dengan Penelitian Sebelumnya	9
2.2 <i>Spesifikasi</i> Sensor MAX9814	12
2.3 <i>Spesifikasi</i> Modul ESP32.....	14
2.4 <i>Spesifikasi DFPlayer</i>	16
4.1 Kalibrasi Sensor Suara.....	39
4.2 Perhitungan Konsumsi Daya.....	41
4.3 Pengujian Web	43
4.4 Monitoring Suara Harian	45


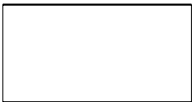
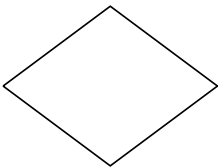
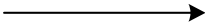
DAFTAR RUMUS

No. Rumus	Halaman
4.1 Error Sensor.....	41
4.2 Rata – Rata Error	41
4.3 Menghitung Daya Alat.....	44
4.4 Rata – rata <i>desibel</i>	45

DAFTAR SINGKATAN

ADC	<i>Analog to Digital Converter</i>
AGC	<i>Automatic Gain Control</i>
CSS	<i>Cascading Style Sheets</i>
DAC	<i>Digital to Analog Converter)</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Domain Name System</i>
FTP	<i>File Transfer Protocol</i>
GND	<i>Ground</i>
GPIO	<i>General Purpose Input/Output</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTML	<i>Hypertext Markup Language</i>
IoT	<i>Internet of Things</i>
RAM	<i>Random Access Memor</i>
PWM	<i>Pulse Width Modulation</i>
RX	<i>Receive</i>
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
TX	<i>Transmit</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
VCC	<i>Voltage at the Common Collector</i>
VCODE	<i>Visual Studio Code</i>

DAFTAR SIMBOL DIAGRAM ALIR

No	Simbol	Nama	Arti Simbol
1		Terminal (<i>Start, End</i>)	Terminal merupakan simbol diagram alir yang menunjukkan awal atau akhir dari suatu kegiatan
2		<i>Processing</i>	<i>Processing</i> merupakan simbol diagram alir yang menunjukkan sebuah proses pengolahan
3		<i>Decision</i>	<i>Decision</i> merupakan simbol diagram alir yang menunjukkan pemilihan atau penentuan suatu keputusan berdasarkan alur yang sedang berjalan
4		<i>Flow Direction</i>	<i>Flow direction</i> merupakan simbol diagram alir yang menjadi penghubung dan penanda arah instruksi selanjutnya

ABSTRAK

Kenyamanan merupakan faktor penting dalam menunjang kegiatan belajar di perpustakaan. Salah satu gangguan utama terhadap kenyamanan tersebut adalah kebisingan yang ditimbulkan oleh aktivitas pengunjung. Untuk mengatasi permasalahan ini, dilakukan perancangan dan pembangunan alat pendeteksi kebisingan berbasis web server yang dapat memberikan peringatan otomatis ketika tingkat suara melebihi ambang batas yang ditentukan. Sistem ini menggunakan sensor suara MAX9814 untuk mendeteksi tingkat kebisingan, ESP32 sebagai mikrokontroler pengolah data, *LCD* sebagai media tampilan, serta modul *DFPlayer* sebagai media peringatan suara. Data *desibel* ditampilkan secara *real-time* melalui web server lokal, lengkap dengan grafik dan *histori* harian yang disimpan di SPIFFS. Sistem juga dilengkapi indikator LED sebagai penanda level kebisingan: hijau (tenang), kuning (berisik), dan merah (sangat berisik). Berdasarkan hasil pengujian, alat mampu mendeteksi tingkat kebisingan dengan rata-rata error sebesar 2,87%. Dan berhasil menampilkan data secara *real-time* baik di *LCD* maupun di website. Sistem ini diharapkan dapat membantu petugas perpustakaan dalam menciptakan lingkungan belajar yang lebih nyaman dan kondusif.

Kata kunci: Kebisingan, ESP32, MAX9814, Web Server, Perpustakaan

ABSTRACT

Comfort is a crucial factor in supporting learning activities in libraries. One of the main disturbances to this comfort is noise generated by visitors. To address this issue, a noise detection and automatic warning system based on a web server has been designed and developed. This system uses the MAX9814 sound sensor to detect noise levels, an ESP32 microcontroller for data processing, an LCD for visual display, and an DFPlayer module for audio warnings. Decibel data is displayed in real time via a local web server, complete with graphical visualization and daily history stored in SPIFFS. The system is also equipped with LED indicators to represent noise levels: green (quiet), yellow (noisy), and red (very noisy). Based on the testing results, the device is capable of detecting noise levels with an average error 2,87% and successfully displays data in real time on both the LCD and the website. This system is expected to assist library staff in creating a more comfortable and conducive learning environment.

Keywords: *Noise, ESP32, MAX9814, Web Server, Library*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perpustakaan adalah salah satu fasilitas atau tempat untuk menggali informasi dan belajar, maka dari itu lingkungan dalam perpustakaan harus dijaga dari faktor-faktor atau hal-hal yang mengganggu kenyamanan, karena kenyamanan sangat berpengaruh dalam menjaga konsentrasi saat melakukan kegiatan belajar agar dapat berjalan secara maksimal. "Kenyamanan merupakan suatu keadaan seseorang merasa sejahtera atau nyaman baik secara mental, fisik maupun sosial."(Keliat, Windarwati, 2015).

Salah satu faktor yang mempengaruhi kurangnya kenyamanan saat belajar terutama saat belajar di perpustakaan adalah kebisingan, kebisingan sendiri adalah suara atau bunyi yang tidak diinginkan yang kebanyakan dihasilkan oleh kegiatan atau aktivitas manusia. Pada ruangan perpustakaan kebisingan masih sering terjadi, yang kebanyakan ditimbulkan oleh pengunjung perpustakaan itu sendiri, hal ini tentunya harus dicegah agar tidak mengganggu kenyamanan bagi pengunjung perpustakaan yang lain. Salah satu hal yang dilakukan untuk mengatasi masalah kebisingan ini adalah diantaranya, dengan memberi peringatan kepada pengunjung yang kedapatan membuat kebisingan oleh petugas perpustakaan, tetapi hal ini tentunya tidak efisien karena petugas perpustakaan tidak bisa selalu mengontrol situasi dalam ruangan perpustakaan agar selalu kondusif, sehingga dibutuhkan sebuah alat otomatis yang dapat mendeteksi kebisingan dan memberi peringatan kepada para pengunjung perpustakaan agar tidak terjadi kebisingan di dalam ruang

perpustakaan.

Berdasarkan latar belakang yang telah diuraikan, maka penulis mengusulkan penelitian dengan judul “Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server”. Yang dimana penelitian ini akan bekerja berbasis *Internet of things (IoT)*, *Internet of Things (IoT)* merupakan sebuah pengembangan komunikasi jaringan dari benda atau alat yang saling terhubung satu sama lain melalui komunikasi internet, yang memungkinkan dapat terjadinya proses pertukaran data informasi secara real time. Sistem IoT pada penelitian ini dimanfaatkan untuk menghubungkan alat pendeteksi kebisingan dengan web server, sehingga proses pendeteksi kebisingan dapat dipantau dan dikontrol melalui web server. Penelitian ini diharapkan dapat membantu petugas perpustakaan untuk menciptakan rasa nyaman di dalam ruang perpustakaan bagi para pengunjung.

1.2. Rumusan Masalah

Berdasarkan penjabaran latar belakang sebelumnya maka rumusan masalah yang dapat digunakan sebagai berikut:

1. Bagaimana cara merancang alat indikator pendeteksi kebisingan di ruang perpustakaan ?
2. Bagaimana sistem kerja dari alat pendeteksi kebisingan di ruang perpustakaan?
3. Bagaimana kinerja dari alat pendeteksi kebisingan di ruang perpustakaan yang telah dibuat ?

1.3. Batasan Masalah

Untuk menghindari meluasnya masalah dalam pembahasan dan perancangan alat ini, maka perlunya dilakukan pembatasan masalah pada tugas akhir ini, adapun batasan masalahnya sebagai berikut:

1. Menggunakan sensor MAX9814 sebagai pendeteksi suara.
2. Menggunakan ESP32 sebagai pengontrol dan pengolah data.
3. Pengujian alat ini dilakukan di ruangan perpustakaan Universitas Tadulako.
4. *Output* kondisi dari alat ini berupa tenang, berisik, dan sangat berisik yang akan ditampilkan pada layar *LCD*

1.4. Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah:

1. Menguji kinerja alat pendeteksi kebisingan pada perpustakaan
2. Mengembangkan sistem pendeteksi kebisingan dengan menggunakan sensor MAX9814 untuk mendeteksi kebisingan
3. Petugas perpustakaan dapat menjaga kenyamanan ruang perpustakaan agar tidak terjadi kebisingan.

1.5. Manfaat Penelitian

Dengan adanya penelitian dan perancangan alat ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Membantu petugas perpustakaan untuk mengawasi para pengunjung perpustakaan agar tidak melakukan tindakan-tindakan yang menimbulkan kebisingan.
2. Membantu menjaga kenyamanan bagi pengunjung ruang perpustakaan.

1.6. Sistematika Penulisan

Sistematika penulisan untuk mendapatkan arah yang tepat mengenai hal-hal yang akan dibahas, maka dalam skripsi ini disusun sebagai berikut:

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Tinjauan pustaka bertujuan untuk mengembangkan pemahaman dan wawasan yang menyeluruh tentang penelitian yang sedang dilakukan, serta berisi informasi-informasi tentang hasil penelitian yang telah dilakukan sebelumnya untuk menjadikan acuan dalam penelitian ini. Landasan teori memuat tentang teori-teori setiap komponen perangkat keras maupun perangkat lunak dalam pembuatan alat peneliti.

BAB III METODE PENELITIAN

Bab ini menjelaskan alat dan bahan yang digunakan dalam penelitian, tahapan-tahapan yang dilakukan selama pelaksanaan penelitian, serta metode penelitian yang disusun berdasarkan teori dasar dan rumusan masalah yang telah ditetapkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil yang diperoleh dari pengujian, analisis, dan pembahasan terkait pengolahan data penelitian, termasuk penyajian dalam bentuk gambar, tabel, dan grafik.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian dan jawaban terhadap rumusan masalah, menegaskan pencapaian tujuan penelitian. Selain itu, bab ini memuat saran-saran untuk penelitian selanjutnya.

DAFTAR PUSTAKA

Berisi tentang daftar judul buku beserta nama penulisnya atau sumber lain yang digunakan penulis saat ini sebagai referensi dalam pelaksanaan penelitian ini.

LAMPIRAN

Berisi dokumentasi penelitian berupa foto proses penelitian dan program yang digunakan pada objek penelitian

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam pembuatan skripsi ini penulis memanfaatkan informasi dari beberapa penelitian sebelumnya sebagai bahan perbandingan baik tentang kekurangan ataupun kelebihan yang sudah ada. Ada beberapa penelitian sebelumnya yang telah dibuat dan berkaitan dengan judul penelitian ini, diantaranya

Dwi Pujiat, dkk, (2022) melakukan penelitian yang berjudul “Perancangan alat Pendeteksi Level Bahaya Kebisingan Area Kerja Berbasis Arduino Uno”. Pada penelitian ini untuk mendeteksi tingkat kebisingannya digunakan sensor suara fc-04 dan menggunakan arduino uno sebagai mikrokontroler untuk mengendalikan rangkaian, untuk sistem kerja dari rangkaian pada penelitian ini sendiri adalah, ketika sensor fc-04 mendeteksi suara data informasi tersebut akan diolah oleh arduino uno yang mana kemudian akan diteruskan ke LED dan *LCD*, apabila tingkat suara yang dideteksi melebihi batas aman untuk didengar manusia maka layar *LCD* akan menampilkan peringatan tanda adanya bahaya kebisingan pada area tersebut.

Syihabuddin, Minto Basuki and Sugiono, (2022) melakukan penelitian yang berjudul “Alat Pendeteksi Kebisingan Di Ruang Pasien Berbasis *Fuzzy Logic*”. Penelitian ini menggunakan sensor KY-038 sebagai sensor pendeteksi suara dan menggunakan mikrokontroler ESP32 sebagai pembaca tingkat kebisingan, pada percobaan ini range kebisingan yang di ukur berkisar antara 30 dB sampai dengan

90 dB, ketika ada suara yang melewati batasan suara yang telah ditentukan yaitu sebesar 75 dB maka akan diberi peringatan menggunakan media *LCD* dan juga buzzer, dan juga akan mengirimkan sebuah notifikasi kepada penjaga ruangan mengenai info kebisingan melalui aplikasi B-lynk.

Widyadana, (2021) melakukan penelitian yang berjudul “Perancangan Sistem Kebisingan Suara Dengan *Sound Sensor Mic* Berbasis Arduino”. Pada percobaan ini untuk mendeteksi suara menggunakan sensor LM393D, sensor suara ini dapat mendeteksi suara yang general yang tingkat batas dari kebisingannya dapat bervariasi seperti contohnya suara tepuk tangan, ketukan benda, dan suara manusia. Sumber suara yang masuk ke sensor ini akan di ubah dari analog ke digital yang kemudian akan ditampilkan pada *LCD*.

Ikhwani, Harmadi and Irfandy, (2021) melakukan penelitian yang berjudul “rancang bangun Alat Peneteksi Kebisingan Berbasis Sensor Serat Optik”. Pada percobaan ini menggunakan sensor serat optik dalam mendeteksi kebisingan, sensor ini dirancang membentuk mikrofon serat optik yang sensitif terhadap bunyi yang diterima, dalam percobaan ini sistem sensor yang dirancang menggunakan serat *optic tipe step-index multimode*, diode laser, dan *fotodetektor* OPT101, Ambang batas tingkat tekanan bunyi yang digunakan dalam percobaan ini adalah 55dB dan peringatannya berupa *display LED dot matrix Max7119*.

Berdasarkan referensi penelitian yang dilakukan oleh beberapa peneliti sebelumnya, maka dilakukan penelitian dengan judul “Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server”. Sistem yang dirancang dalam percobaan ini menggunakan sensor MAX9814 sebagai sensor

untuk mendeteksi kebisingan atau suara, yang kemudian hasil dari pembacaan sensor suara tersebut akan dibagi menjadi tiga tingkatan atau level berbeda yaitu pada level pertama 0dB-50dB yang dimana masuk dalam tingkatan kategori tenang, kemudian 51dB-70dB yang masuk dalam tingkatan kategori berisik, dan >70dB yang masuk dalam tingkatan kategori sangat berisik.

Tiga tingkatan kebisingan ini akan di tandai dengan 3 *LED* yang memiliki warna berbeda, yang dimana untuk kondisi tenang atau 0dB-50dB *LED* yang akan menyala atau terpicu adalah *LED* berwarna hijau dan tidak akan diberi peringatan apapun. Kemudian ketika sensor mendeteksi suara pada *range* 51dB-70dB atau kondisi berisik maka *LED* yang akan menyala adalah *LED* berwarna kuning dan akan diberi sebuah peringatan yang ditampilkan pada *liquid crystal display (LCD)* dan pemberitahuan melalui audio agar pengunjung perpustakaan dapat kembali tenang, dan selanjutnya saat sensor mendeteksi suara pada *range* >70dB maka lampu *LED* yang akan menyala adalah *LED* yang berwarna merah yang kemudian akan diberikan peringatan melalui *LCD* dan juga akan memicu terputarnya pesan peringatan audio dari modul *DFPlayer* agar para pengunjung perpustakaan dapat tenang dan tidak menimbulkan kebisingan yang dapat mengganggu pengunjung lainnya.

Untuk pengolahan data dari pembacaan sensor suara itu sendiri digunakan mikrokontroler ESP32 sebagai otak dari perancangan alat ini, dan hasil pengolahan data dari ESP32 akan ditampilkan dalam sebuah halaman *web*, dalam *web* tersebut hasil pembacaan sensor MAX9814 dapat dipantau dan pengawas perpustakaan dapat juga mengontrol alat secara manual melalui *web*, baik itu untuk mengaktifkan

ataupun menonaktifkan alat.

Tabel 2. 1 Perbandingan dengan penelitian sebelumnya

Dwi Pujiat dkk (2022) , “Perancangan Alat Pendeteksi Level Bahaya Kebisingan Area Kerja Berbasis Arduino Uno”		
Persamaan	Perbedaan	
	Penelitian Terdahulu	Penelitian Penulis
1. Peringatan Menggunakan <i>LCD</i> . 2. Menggunakan <i>LED</i> sebagai Indikator.	1. Menggunakan sensor suara fc-04 2. Menggunakan Arduino Uno. 3. Peringatan hanya menggunakan <i>LCD</i> .	1. Menggunakan sensor MAX9814 sebagai sensor pendeteksi suara. 2. Menggunakan ESP32. 3. Peringatan menggunakan <i>LCD</i> dan Audio.
Achmad Syihabuddin (2022), “Alat Pendeteksi Kebisingan Di Ruang Pasien Berbasis <i>Fuzzy Logic</i>”		
Persamaan	Perbedaan	
	Penelitian Terdahulu	Penelitian Penulis
1. Menggunakan ESP32 2. Pemberi peringatan menggunakan <i>LCD</i>	1. Menggunakan algoritma fuzzy 2. Tidak memiliki level kebisingan. 3. Peringatan suara audio menggunakan buzzer 4. Menggunakan Sensor KY-038	1. Tidak menggunakan algoritma. 2. Memiliki level kebisingan 3. Peringatan <i>audio</i> menggunakan modul ISD1820. 4. Menggunakan sensor MAX9814
Theodorus S Kalengkongan (2018), “Perancangan Sistem Kebisingan Suara Dengan <i>Sound Sensor Mic</i> Berbasis Arduino”		
Persamaan	Perbedaan	
	Penelitian Terdahulu	Penelitian Penulis
1. Penelitian tentang pendeteksi tingkat kebisingan	1. Memakai sensor LM393D sebagai sensor suara	1. Memakai sensor MAX9814 sebagai sensor suara.

2. Memiliki tingkat level kebisingan	2. Memberi peringatan hanya menggunakan <i>LCD</i> . 3. Menggunakan mikrokontroler arduino.	2. Memberi peringatan menggunakan <i>LCD</i> dan Audio. 3. Menggunakan mikrokontroler ESP32.
Sixtinah Deswila (2019), “Rancang bangun Alat Pendeteksi Kebisingan Berbasis Sensor Serat Optik”		
Persamaan	Perbedaan	
	Penelitian Terdahulu	Penelitian Penulis
Penelitian tentang pendeteksi kebisingan	1. Menggunakan sensor serat optik dalam mendeteksi suara 2. Menggunakan mikrokontroler Arduino 3. <i>Output</i> berupa tampilan <i>LCD</i> OLED.	1. Menggunakan sensor MAX9814 sebagai sensor suara 2. Menggunakan mikrokontroler ESP32. 3. <i>Output</i> berupa tampilan <i>LCD</i> dan <i>Audio</i> .

2.2 Landasan Teori

Landasan teori pada penelitian ini membahas konsep-konsep dasar yang mendukung pengembangan Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web. Pembahasan mencakup teori dari penelitian sebelumnya mengenai komponen yang digunakan didalam penelitian ini.

2.2.1 Kebisingan

Kebisingan atau *Noise* merupakan bunyi yang tidak dikehendaki (Aperti 2018),. Kebisingan berdasarkan pengaruhnya kepada manusia dapat dibedakan menjadi tiga jenis yaitu: bising yang mengganggu (*Irritating noise*), bising yang

menutupi (*Masking noise*) dan bising yang merusak (*Damaging /Injurious noise*).

Terdapat beberapa faktor penyebab kebisingan, yaitu :

1. Tekanan Suara

Tekanan suara merupakan satuan daya tekanan suara per satuan luas.

Bunyi akan mengadakan suatu penekanan ketika melalui sebuah medium rambat.

2. Daya Suara

Daya suara atau disebut juga daya akustik merupakan energi bunyi yang dikeluarkan atau dipancarkan oleh suatu sumber bunyi per satuan waktu, dan mempunyai satuan *Joule/s* atau *Watt*. Daya suara tidak dipengaruhi oleh jarak.

3. Intensitas Suara

Intensitas suara merupakan energi rata-rata dari suara yang ditransmisikan oleh gelombang suara menuju arah rambat media. Intensitas suara sangat dipengaruhi oleh jarak. Semakin jauh dari sumber bunyi atau semakin besar luasan yang ditembus maka intensitas suaranya semakin kecil.

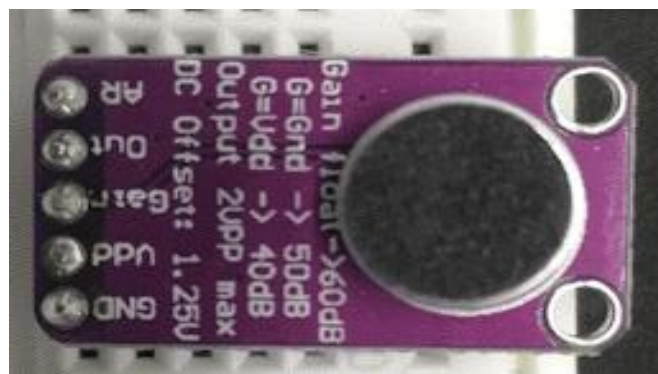
4. Frekuensi

Frekuensi merupakan getaran yang dihasilkan dalam satuan waktu (detik) dengan satuan Hz. Rata-rata manusia dapat mendengar frekuensi di kisaran 20 Hz sampai 20000 Hz, dan saat berbicara manusia dapat menghasilkan nilai frekuensi antara 125 Hz-2000 Hz. Nilai frekuensi dapat dihitung dengan menggunakan rumus :

2.2.2 Sensor MAX 9814

Sensor suara MAX9814 adalah modul mikrofon *electret* dengan penguat

otomatis (AGC - *Automatic Gain Control*) yang dirancang untuk menangkap suara dengan sensitivitas tinggi dan kejernihan yang baik, terutama dalam lingkungan dengan tingkat kebisingan yang berubah-ubah. Modul ini sangat populer dalam aplikasi sistem pengenalan suara, pengukuran tingkat kebisingan (*desibel*), serta sistem monitoring audio berbasis mikrokontroler seperti ESP32. Adapun tampilan fisik dari sensor MAX9814 seperti pada gambar 2.2.



Gambar 2. 2 Sensor MAX 981

Modul MAX9814 terdiri dari tiga komponen utama:

1. Mikrofon *electret condenser*: Menangkap gelombang suara dan mengubahnya menjadi sinyal listrik.
2. Penguat operasional (Op-Amp): Memperkuat sinyal *analog* dari mikrofon.
3. Sirkuit AGC (*Automatic Gain Control*): Mengatur penguatan (*gain*) secara otomatis agar output tetap stabil, meskipun intensitas suara berubah-ubah.

AGC sangat berguna karena mencegah distorsi pada suara keras dan tetap menjaga sensitivitas untuk suara pelan. Hal ini memungkinkan pembacaan suara yang lebih konsisten dalam berbagai kondisi. Adapun *spesifikasi* dari sensor MAX9814 terdapat pada tabel 2.2 :

Tabel 2.2 Spesifikasi Sensor MAX9814

Fitur	<i>Spesifikasi</i>
Tegangan Operasi	2.7V – 5.5V
Tipe Mikrofon	<i>Electret Condenser</i>
Output	Analog (bisa langsung dibaca oleh ADC mikrokontroler)
Fitur Tambahan	AGC (<i>Automatic Gain Control</i>), tiga pilihan penguatan
Rentang Frekuensi	20Hz – 20kHz (rentang pendengaran manusia)
Waktu Serang AGC	Cepat (FAST), Menengah (MED), Lambat (SLOW)
Ukuran Modul	±20mm x 15mm
Konsumsi Arus	±1 mA
Output Noise	Rendah (low-noise amplifier)
Kontrol Gain	External pin untuk memilih penguatan 40dB, 50dB, 60dB

(sumber : <https://www.analog.com/media/en/data-sheets/max9814.pdf>)

2.2.3 ESP32

ESP32 adalah sebuah modul *mikrokontroler* terintegrasi yang memiliki fitur cukup lengkap dan memiliki kinerja tinggi. ESP32 memiliki dua prosesor komputasi, yang dimana satu digunakan untuk mengelola jaringan *wifi* dan *Bluetooth*, dan satunya digunakan untuk menjalankan aplikasi, *Mikrokontroler* ini juga memiliki penyimpanan memori *RAM* yang cukup besar untuk menyimpan

data.

ESP32 sangat cocok untuk digunakan dalam proyek-proyek *Iot (Internet of Things)*. Karena *mikrokontroler* ini mampu menghubungkan perangkat ke jaringan internet dengan mudah. *Mikrokontroler* ini dapat terhubung ke jaringan *WiFi* dengan menggunakan *protocol* TCP/IP yang memungkinkan ESP32 berkomunikasi dengan perangkat lain. Dan juga, ESP32 mempunyai fitur *Bluetooth* untuk menghubungkan perangkat lain. ESP32 juga memiliki banyak keunggulan lain seperti kemampuan *multitasking* yang luar biasa, konsumsi daya yang rendah, dan harga yang terjangkau, adapun spesifikasi dari ESP32 dapat dilihat pada tabel 2.3.

Tabel 2.3 Spesifikasi ESP32

Spesifikasi	Keterangan
Microprosesor	Xtensa Dual-Core 32 Bit LX6
<i>Freq Clock</i>	up to 240 MHz
SRAM	520 kB
<i>Flash memori</i>	4 MB
WiFi <i>transceiver</i>	11b/g/n
Bluetooth	4.2/BLE
pin GPIO	48 pin GPIO
Pin channel ADC (<i>Analog to Digital Converter</i>)	15 pin channel ADC (<i>Analog to Digital Converter</i>)
Pin PWM (<i>Pulse Width Modulation</i>)	25 pin PWM (<i>Pulse Width Modulation</i>)
Pin channel DAC (<i>Digital to Analog Converter</i>)	2 pin channel DAC (<i>Digital to Analog Converter</i>)

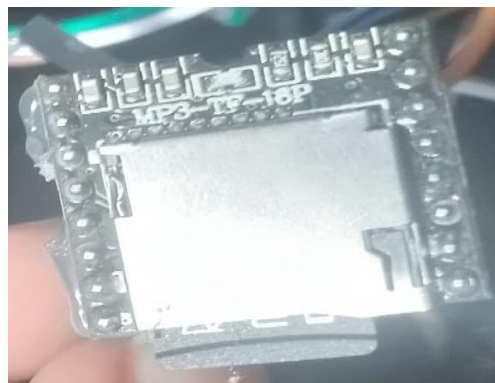
(sumber : <https://www.ardutech.com>)



Gambar 2. 3 ESP32

2.2.4 DFPlayer

DFPlayer Mini adalah sebuah modul pemutar audio berukuran kecil yang dirancang untuk memutar *file audio digital* (seperti MP3, WAV, dan WMA) dari kartu microSD. Modul ini sangat cocok digunakan dalam berbagai proyek berbasis mikrokontroler seperti ESP32, terutama pada sistem yang membutuhkan audio output atau peringatan suara otomatis. *DFPlayer* Mini mampu bekerja secara mandiri maupun terintegrasi dengan mikrokontroler melalui komunikasi serial UART. Bentuk fisik dan spesifikasi dari DFPlayer dapat dilihat pada gambar 2.4 dan tabel 2.4.



Gambar 2. 4 DFPlayer

Modul suara *DFPlayer* mempunyai beberapa kelebihan dan kekurangan yaitu:

Kelebihan DFPlayer :**1. Mudah digunakan**

DFPlayer Mini dirancang dengan antarmuka yang sederhana, sehingga mudah diintegrasikan ke berbagai sistem mikrokontroler seperti Arduino dan ESP32. Modul ini juga dapat bekerja secara mandiri (standalone) menggunakan tombol tekan tanpa mikrokontroler.

2. Mendukung komunikasi serial UART

Modul ini dapat dikendalikan melalui komunikasi serial UART, memungkinkan pengguna untuk mengatur pemutaran file, volume, dan mode langsung melalui kode program.

3. Mendukung berbagai format file audio

DFPlayer Mini kompatibel dengan format audio populer seperti MP3, WAV, dan WMA, yang membuatnya fleksibel dalam berbagai kebutuhan audio sederhana.

4. Memiliki dua jenis output audio

DFPlayer Mini menyediakan output DAC (line out) untuk koneksi ke amplifier eksternal dan output amplifier internal (melalui pin SPK_1 dan SPK_2) untuk langsung terhubung ke speaker kecil.

5. Dapat digunakan dengan kartu *microSD* hingga 32GB

Modul ini mendukung kartu *microSD* berkapasitas besar (hingga 32GB) dengan format FAT16 atau FAT32, memungkinkan penyimpanan banyak file audio.

6. Ukuran kecil dan hemat daya

Dengan dimensi yang ringkas dan konsumsi daya rendah, *DFPlayer* Mini sangat cocok digunakan dalam proyek embedded system atau perangkat portabel bertenaga baterai.

7. Harga terjangkau dan mudah diperoleh

DFPlayer Mini tersedia luas di pasaran dengan harga yang relatif murah, menjadikannya solusi audio yang efisien untuk berbagai proyek DIY dan edukasi.

Kekurangan:

1. Minimnya umpan balik status pemutaran

Modul ini tidak memberikan informasi langsung kepada mikrokontroler apakah file berhasil diputar atau telah selesai, kecuali dengan memanfaatkan pin *BUSY* secara manual.

2. Terbatas pada format sistem file tertentu

DFPlayer Mini hanya mendukung kartu *microSD* yang diformat dengan sistem file FAT16 atau FAT32, sehingga tidak kompatibel dengan format exFAT yang digunakan pada kartu berkapasitas lebih besar.

3. Struktur folder dan penamaan file terbatas

Modul ini hanya dapat membaca folder yang dinamai dengan angka (misalnya “01”, “02”, dst.), dan file audio harus diberi nama sesuai format numerik seperti “001.mp3”, “002.mp3”, dan seterusnya.

4. Kualitas output audio internal terbatas

Meskipun dapat langsung menggerakkan speaker kecil, kualitas suara dari amplifier internal relatif rendah. Untuk hasil audio yang lebih jernih dan keras, dibutuhkan amplifier eksternal.

5. Kontrol audio terbatas

Fungsi kendali audio melalui perintah serial cukup terbatas. Modul ini tidak menyediakan fitur seperti pencarian (seek), pengulangan spesifik, atau navigasi playlist dinamis yang lebih kompleks.

Tabel 2.4 Spesifikasi DFPlayer

Spesifikasi	Keterangan
Tegangan Operasi	3.2V – 5.0V DC
Format File Audio	MP3, WAV, WMA
Kapasitas Kartu microSD	Maksimal 32 GB (format FAT/FAT32)
Komunikasi	Serial UART (9600 bps default)
Output Audio	DAC_L / DAC_R (line out), SPK_1 / SPK_2 (amplifier out)
Daya Output Speaker	Maksimal 3W (melalui amplifier internal)
Jumlah Maksimum File Audio	Hingga 255 file
Mode Operasi	Serial UART atau Standalone (dengan tombol)
Konsumsi Arus	± 20 mA saat idle, lebih tinggi saat memutar audio
Ukuran Modul	± 22 mm x 30 mm

(Sumber : <https://www.alldatasheet.com/html/ETC2/DFPlayer.html>)

2.2.5 Liquid Crystal Display (LCD)

Liquid crystal display (LCD) adalah salah satu jenis layar panel datar yang menggunakan kristal cair dalam pengoperasiannya, *LCD* sudah digunakan diberbagai bidang misalnya dalam alat-alat elektronika seperti televisi, kalkulator, maupun layar komputer/laptop, bentuk fisik dari LCD dapat dilihat pada gambar 2.5. Terkhusus pada penelitian ini layar *LCD* akan digunakan untuk memberi peringatan pemberitahuan kepada para pengunjung perpustakaan untuk tidak menimbulkan kebisingan di dalam ruang perpustakaan, yang dimana layar *LCD* akan diprogram untuk menampilkan sebuah kalimat pemberitahuan agar dapat

dilihat oleh para pengunjung perpustakaan. Cara kerja *LCD*:

1. Lapisan Kristal Cair: *LCD* terdiri dari dua lapisan kaca atau plastik dengan lapisan tipis kristal cair di antaranya. Kristal cair ini dapat berubah orientasi saat terkena medan listrik, yang memungkinkan atau menghalangi cahaya melewatinya.
2. Sumber Cahaya: *LCD* memerlukan sumber cahaya dari belakang (backlight), karena kristal cair sendiri tidak menghasilkan cahaya. Sumber cahaya ini biasanya berupa *LED (Light Emitting Diode)* atau lampu fluoresen.
3. Pixel: Setiap piksel di layar *LCD* terdiri dari tiga subpiksel dengan warna merah, hijau, dan biru (RGB). Dengan mengontrol intensitas cahaya yang melewati subpiksel ini, layar dapat menampilkan berbagai warna dan gambar.
4. Polarisasi Cahaya: Ketika medan listrik diterapkan ke kristal cair, molekul-molekul kristal ini memutar polarisasi cahaya yang melewatinya. Cahaya ini kemudian melewati filter polarisasi, yang menentukan apakah cahaya tersebut akan mencapai permukaan layar atau tidak, sehingga menciptakan gambar



Gambar 2. 5 *Liquid Crystal Display*

2.2.6 *Light Emitting Diode (LED)*

Light Emitting Diode (LED) merupakan salah satu dari jenis perangkat

semikonduktor yang mengeluarkan cahaya ketika di aliri arus listrik, berbeda dengan lampu pijar yang menggunakan fillamen atau gas untuk menghasilkan cahaya, *LED* bekerja menggunakan prinsip elektroluminesensi, yang dimana cahaya *LED* dihasilkan dari pergerakan elektron di dalam material semikonduktor. Pada penelitian ini *LED* dimanfaatkan sebagai indikator pemberitahuan level kebisingan yang dihasilkan dalam ruang perpustakaan, yang dimana digunakan 3 warna *LED* yang berbeda yang masing-masing mewakili *range* suara yang diterima oleh sensor, yang pertama adalah warna hijau yang mewakili *range* suara 0 – 50 dB, kemudian kuning yang mewakili *range* suara 51 – 70 dB, dan yang terakhir merah yang mewakili *range* suara >71 dB. Bentuk fisik LED dapat dilihat pada gambar 2.6.



Gambar 2. 6 *Light Emitting Diode*

2.2.7 Web Server

Web server adalah perangkat lunak yang memberi layanan berupa data. Web server bertugas untuk menerima permintaan HTTPS atau HTTP atau bisa dikenal dengan web browser seperti chrome dan firefox dari pengguna internet. Setelah itu, web server akan menyediakan respons atas permintaan tersebut dalam bentuk

halaman web. Web server bekerja dalam beberapa tahapan, yaitu :

Pertama, pengguna akan memasukan permintaan melalui web *browser*, seperti mengunjungi sebuah *website* ataupun mengirim sebuah email. Lalu *browser* akan meminta data yang dibutuhkan melalui HTTP/HTTPS, HTTP/HTTPS merupakan protokol untuk mendukung proses pemindahan data informasi melalui gambar, teks, atau komponen lainnya.

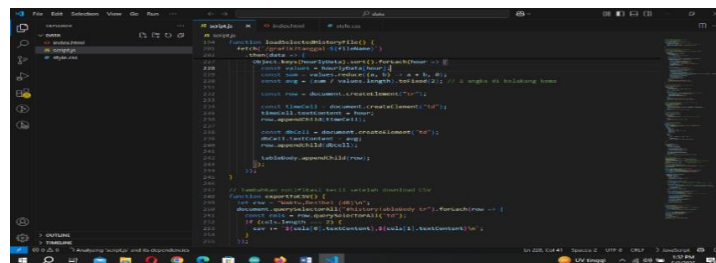
Kedua, web server akan mencari permintaan browser dalam hosting. Hosting adalah tempat untuk menyimpan file-file yang dibutuhkan *website*. Ketika data-data yang dicari sudah ditemukan, web server akan menampilkan halaman *error*. Terakhir, browser akan menampilkan informasi yang diminta kepada *user*.

Web server memiliki banyak jenis yang berbeda-beda, dengan kelebihan dan kekurangannya masing-masing, jenis-jenis nya yaitu antara lain:

1. Nginx, adalah sebuah web server *open source* yang artinya bisa digunakan oleh siapa saja dengan bebas tanpa membayar. Web server ini memiliki keunggulan Apache dari segi penanganan website dengan *traffic* yang tinggi.
2. Apache, adalah sebuah web server yang paling populer dan sudah banyak digunakan oleh masyarakat. Fungsi dari apache adalah untuk menggantikan fungsi dari server, apache mendukung beberapa program yang melengkapinya, diantaranya adalah PHP, SSI, dan *Access Control*.
3. IIS, adalah sebuah web server yang mendukung jenis protokol DS, TCP/IP, atau server lainnya yang dipakai untuk merancang situs.

2.2.9 Visual Studio Code

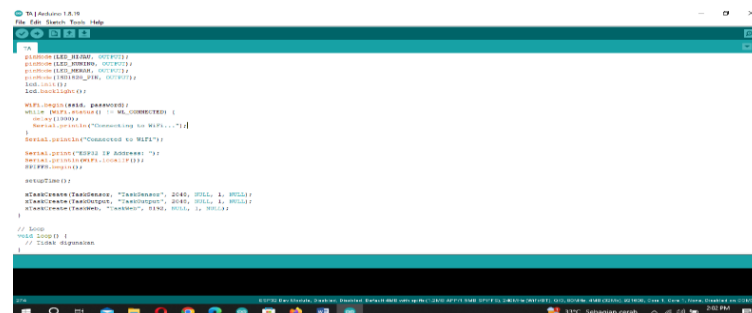
Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node. Js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang *via marketplace Visual Studio Code* seperti : C++, C#, Python, Go, Java, PHP, dst. (Ummy Gusti Salamah, 2021).



Gambar 2. 8 Visual Studio Code

2.2.10 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah *software* yang digunakan untuk membuat *sketch* pemrograman atau kode pemrograman, dengan kata lain Arduino IDE adalah sebuah media untuk pemrograman pada board yang ingin diprogram.



Gambar 2. 9 Arduino IDE

Arduino IDE berguna untuk mengedit, membuat, meng-upload program ke *board*. Arduino IDE menggunakan Bahasa dari pemrograman JAVA, dan dilengkapi dengan *library* C/C++, yang membuat operasi *input/output*nya menjadi lebih mudah.

2.2.11 Desibel

Desibel (dB) secara sederhana adalah satuan logaritmik yang digunakan untuk mengukur seberapa keras atau lembut suara dibandingkan ambang dengar manusia, Leslie L. Doelle (1986) mengatakan bahwa “ *Desibel* (dB) adalah perubahan terkecil dalam tekanan bunyi yang dapat dideteksi telinga pada umumnya”.

Tingkatan tekanan bunyi : Kantor pribadi, rumah yang tenang, percakapan yang tenang : 20– 40 dB (lemah). Rumah yang bising, percakapan pada umumnya : 40 – 60 Db (sedang). Kantor yang bising : 60 – 80 dB (keras). Bising lalu lintas : 80 – 100 dB (sangat keras). James Cowan (2010) mengatakan bahwa : *Desibel* (dB) adalah ukuran kekuatan medan bunyi pada skala logaritmik. Dapat digunakan untuk menunjukkan besarnya tingkat bunyi pada suatu titik dalam sebuah medan bunyi atau jumlah keseluruhan tingkat kekuatan sebuah sumber bunyi. Dapat didefinisikan secara matematik sebagai 10 dikalikan dengan logaritma dari kuantitas yang diukur dengan nilai referensi dari kuantitas yang sama, dimana kuantitas berhubungan dengan kekuatan dari sumber.

Desibel merupakan satuan yang sering digunakan sebagai skala penguatan dalam rangkaian Elektronika seperti rangkaian pada peralatan *Audio* dan Komunikasi. Besaran-besaran yang menggunakan skala penguatan *Desibel* tersebut diantara seperti penguatan pada Daya, Tegangan, Arus dan juga Intensitas suara.

Jadi pada dasarnya *Desibel* adalah satuan yang menggambarkan suatu perbandingan atau Rasio. Secara definisi, *Desibel* yang sering disingkat dengan “dB” ini dapat diartikan sebagai “Perbandingan antara dua besaran dalam skala Logaritma”.

BAB III

METODE PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan penelitian rancang bangun alat pendeteksi kebisingan ini akan diuraikan di bawah ini:

3.1.1 Alat

Adapun alat yang akan digunakan dalam penelitian ini terbagi atas dua jenis yaitu perangkat keras dan perangkat lunak.

a. Perangkat Keras

1. Laptop
2. Solder
3. *Mikrokontroler* ESP32
4. Sensor Max 9814
5. *DFPlayer*
6. LCD
7. *Speaker*

b. Perangkat Lunak (*Software*)

1. Arduino IDE
2. Fritzing
3. *Visual Studio Code*

3.1.2 Bahan

1. Kabel jumper
2. Timah
3. Lem tembak

3.2 Waktu Dan Lokasi Penelitian

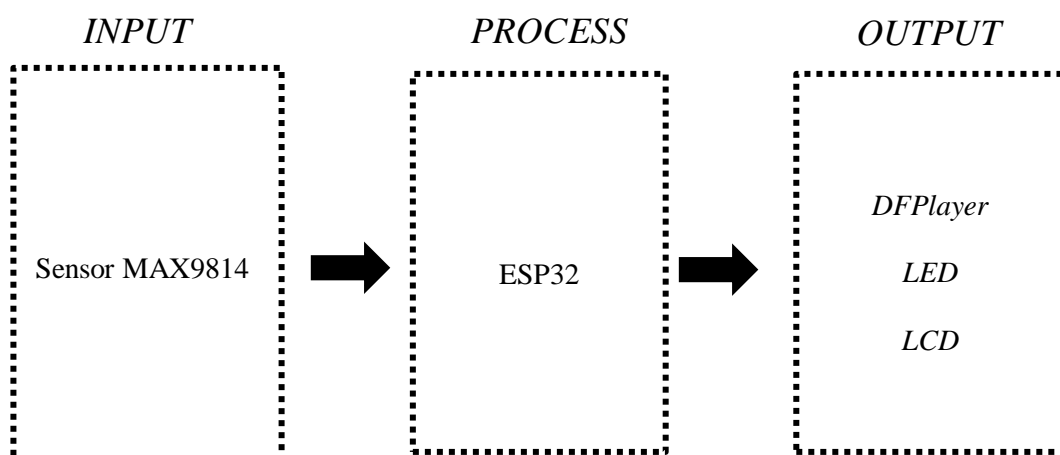
Penelitian Sistem Pendeteksi Suara Dan Pemberi Peringatan Di Perpustakaan Berbasis *Web Server* yang meliputi pengujian alat dilakukan di Perpustakaan Universitas Tadulako. Penelitian ini dilakukan dalam jangka waktu 6 bulan, yakni bulan November 2024 sampai Mei 2025.

3.3 Tahapan Penelitian

Tahapan penelitian ini dilakukan untuk memastikan pengembangan sistem deteksi kebisingan dan pemberi peringatan berbasis *web server* berjalan sesuai dengan yang diinginkan.

3.3.1 Diagram Blok

Adapun diagram blok rancangan sistem terdapat pada gambar 3.2 berikut



Gambar 3. 2 Diagram Blok Perancangan Sistem

Dalam merancang Sistem Pendeteksi Suara Dan Pemberi Peringatan Di Perpustakaan Berbasis *Web Server* memiliki 3 bagian penting yaitu:

1. Sistem *input*

Ada dua komponen yang berperan sebagai input, yaitu sensor MAX9814. Sensor MAX9814 digunakan untuk mendeteksi suara yang menyebabkan kebisingan di dalam ruangan perpustakaan, suara yang diterima oleh sensor akan dikirimkan ke mikrokontroler untuk diproses lebih lanjut.

2. Sistem proses

Pada sistem proses, ESP32 berfungsi sebagai pengendali utama yang memproses data dari sensor MAX9814 dan sebagai pengirim data ke web server. Ketika sensor MAX9814 mendeteksi suara, data tersebut dikirimkan ke ESP32. ESP32 kemudian memproses data tersebut, apakah data suara tersebut masuk kedalam kategori tenang, bising, dan sangat bising. ESP32 juga dapat mengirimkan data ini melalui koneksi Wi-Fi ke server web.

3. Sistem *output*

Pada sistem ini, terdapat beberapa *output* yang digunakan, yaitu *LED*, *LCD*, dan *DFPlayer*. *LED* berfungsi sebagai indikator kondisi dalam ruangan perpustakaan, ada tiga buah *LED* yang digunakan dalam sistem ini yaitu *LED* hijau sebagai indikator untuk kondisi tenang, *LED* kuning untuk indikator kondisi berisik, dan *LED* merah untuk indikator untuk kondisi sangat berisik. *LCD* untuk menampilkan kondisi dalam ruangan dan menampilkan halaman website. Dan *DFPlayer* digunakan untuk memberikan peringatan kepada pengunjung untuk tenang ketika sistem mendeteksi desibel suara diatas 50dB.

3.3.2 Diagram Alir

Adapun diagram alir tahapan penelitian Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server yang akan dilakukan terdapat pada gambar 3.1.

1. Mulai

Mulai merupakan awal dari pembuatan sistem pendeteksi suara dan pemberi peringatan berbasis *web server*.

2. Studi literatur

Pada tahap ini dilakukan pencarian referensi yang sesuai mengenai yang kiranya dapat membantu dalam penelitian ini.

3. Perancangan *Hardware* dan *Software*

Pada tahap ini dilakukan perancangan terhadap komponen yang akan digunakan berupa pengecekan sensor MAX 9814 untuk mendeteksi suara secara *real time*, pengecekan *LCD* dan *DfPlayer* dalam memberi pemberitahuan. ESP32 akan digunakan sebagai *mikrokontroler* dan seluruh data akan ditampilkan lewat *website*.

4. Perakitan alat

Pada tahap ini dilakukan perakitan komponen sesuai dengan skematik yang telah dirancang sehingga semua komponen yang digunakan dapat terhubung dengan ESP32.

5. Pembuatan program ESP32

Pada tahap ini dilakukan pemrograman pada ESP32 menggunakan Bahasa C++ pada *software* Arduino IDE.

6. Pengujian alat

Pada tahap ini dilakukan Pengujian alat untuk mengetahui apakah alat sudah berfungsi dengan baik sesuai dengan program yang diinginkan serta mengecek *output* pada sensor dalam mendeteksi suara. Jika pengujian ini tidak sesuai dengan apa yang diinginkan maka akan dilakukan perakitan dan pemrogram kembali.

7. Pembuatan Website

Pada tahap ini dilakukan pembuatan *website* dengan menggunakan *software visual studio code* dengan menggunakan bahasa php serta membuat struktur *website* menggunakan HTML untuk konten, CSS untuk desain, dan javascript untuk menghubungkan tampilan web dengan esp32

8. Pembuatan program untuk menghubungkan ESP32 ke website

Tahap ini dilakukan untuk menghubungkan ESP32 ke *website* agar data dari sensor dapat ditampilkan ke *website*.

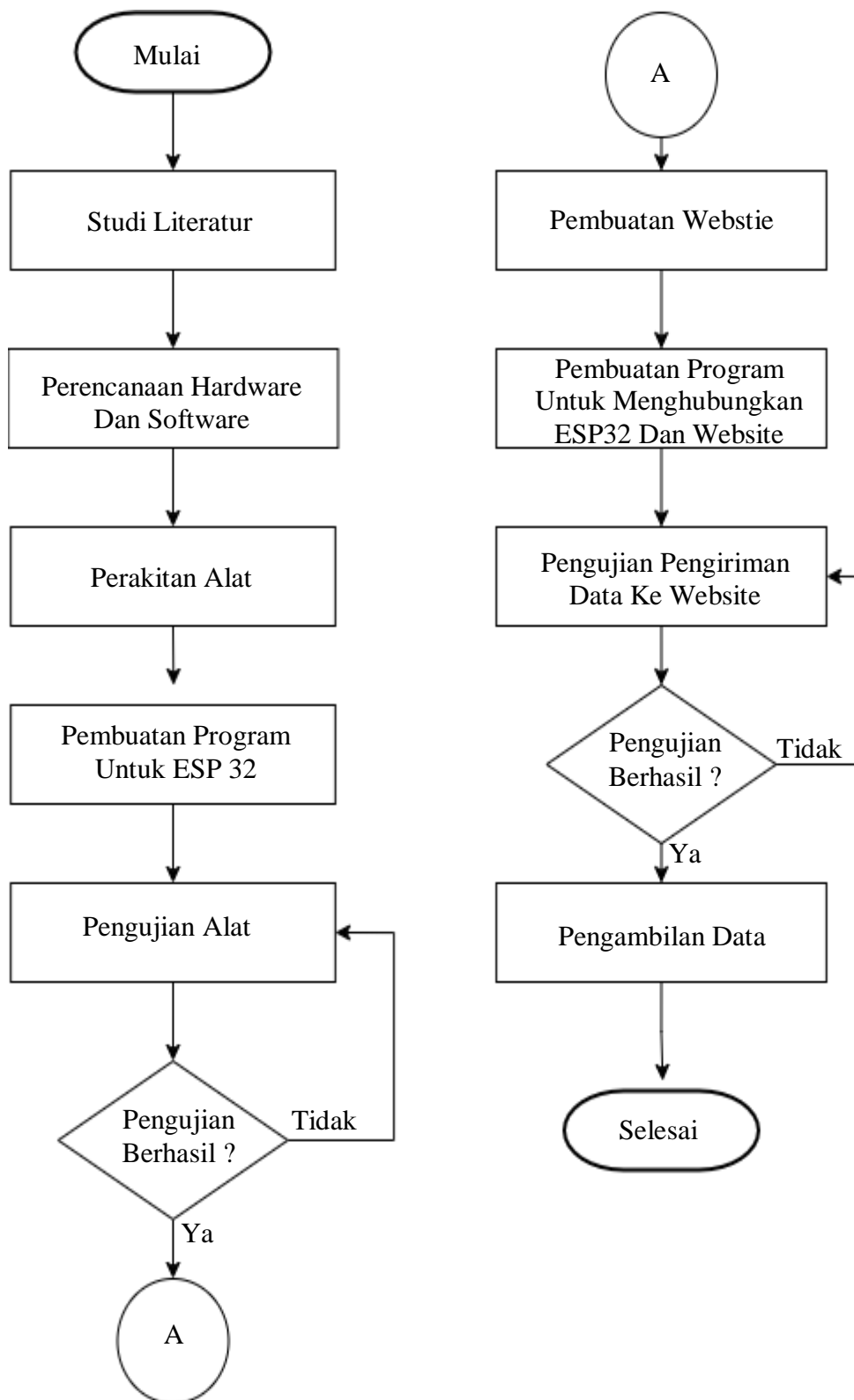
9. Pengujian pengiriman data ke website

Pengujian ini dilakukan untuk mengetahui apakah data dari ESP32 dapat terlihat di *website*. Apabila pengujian ini tidak sesuai dengan apa yang diinginkan maka akan dilakukan perakitan dan pemrogram kembali.

10. Pengambilan data

Langkah ini dilakukan bila semua langkah-langkah telah memenuhi syarat. Data yang diambil berupa *output* dari sensor MAX9814 berupa besaran *desibel* suara.

11. Selesai



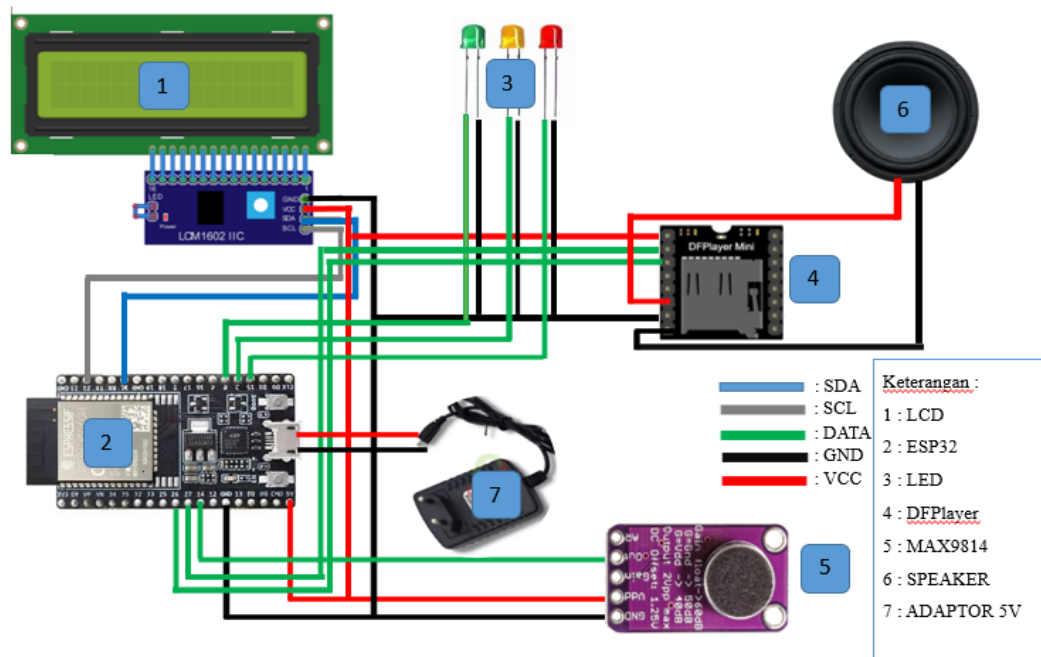
Gambar 3.1 Diagram Alir Penelitian

3.3.3 Perancangan Sistem

Rancangan sistem dilakukan untuk pembuatan Sistem Deteksi suara dan pemberi peringatan di perpustakaan berbasis web. Tahap ini mencakup skema rancangan sistem, diagram alir prinsip kerja alat , tampilan halaman *web*, serta tampilan alat.

A. Skema Rancangan Sistem

Adapun skematik perancangan sistem terdapat pada gambar 3.3 berikut ini



Gambar 3. 3 Skema Rancangan Sistem

Rancangan rangkaian sistem dari simulasi alat pendeteksi kebisingan suara dan pemberi peringatan di perpustakaan universitas tadulako, yang dimana alat ini terdiri dari *LCD* yang sudah terhubung dengan I2C dimana I2C mempunyai pin SDA dan pin SCL yang dihubungkan melalui pin 21 dan 22 di ESP32, VCC ke pin 5V dan dan GND ke pin GND yang juga terhubung ke GND sensor suara

MAX9814, *VCC* pada sensor suara terhubung ke pin 5v, dan pin *OUT* terhubung ke pin 32 pada ESP32. Kemudian modul *DFPlayer* port *VCC* terhubung ke pin 5V, *GND* terhubung ke pin *GND* pada ESP32, dan pin RX dan TX terhubung ke pin 37 dan 36 pada ESP32 untuk menjalankan modul suara.

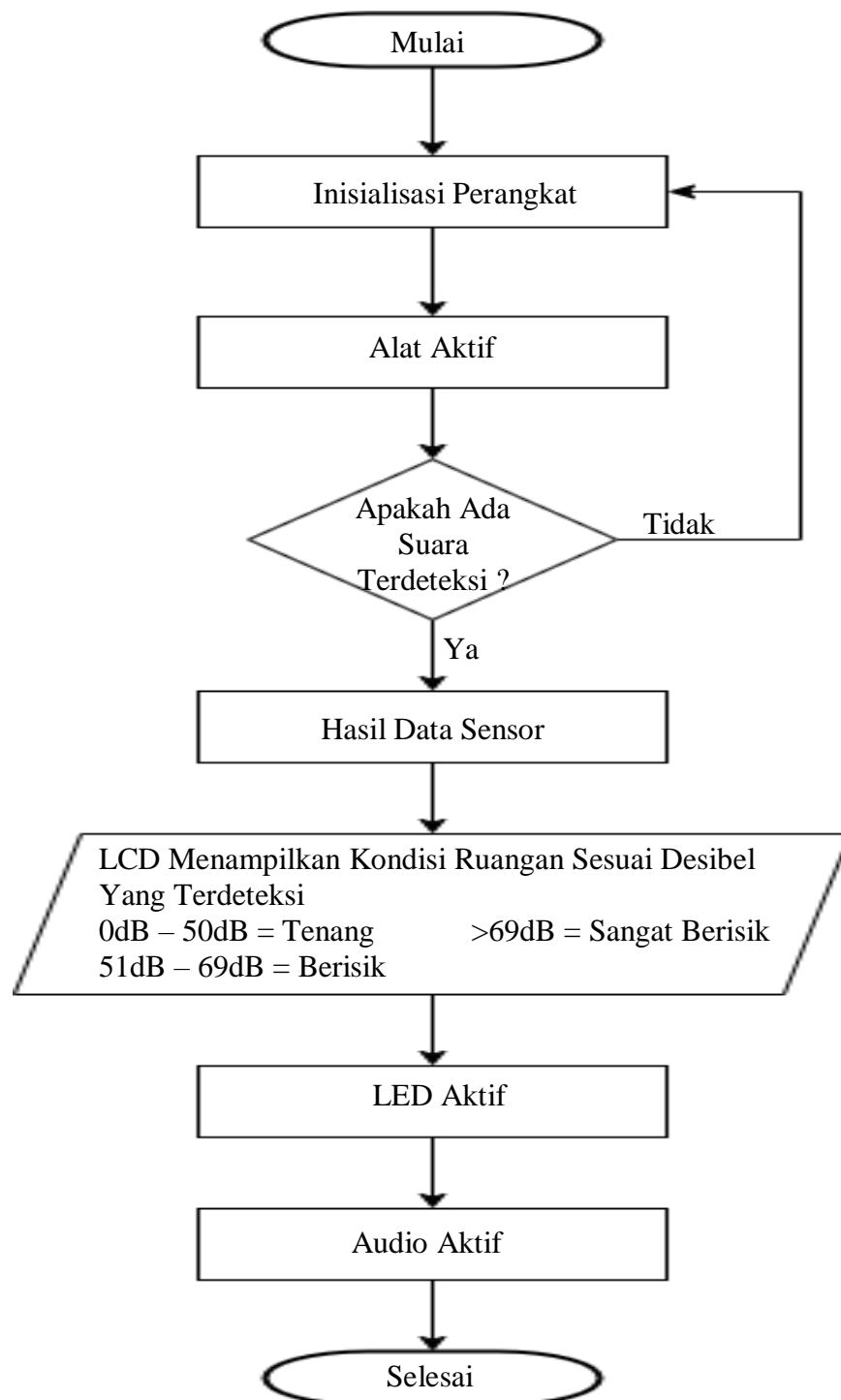
B. Diagram Alir Prinsip Kerja Alat

Pada diagram alir prinsip kerja sensor yang terdapat pada gambar 3.4 Pada prinsip kerja alat, yang pertama dilakukan adalah mengaktifkan alat dan inisialisasi serta mengkonfigurasi seluruh rangkaian sistem yang saling terhubung, kemudian sensor akan mendeteksi suara yang ada di dalam ruangan perpustakaan, jika sensor berhasil mendeteksi suara maka hasil data sensor akan diproses oleh ESP32 yang kemudian akan diklasifikasikan menjadi tiga kondisi seperti pada tabel 3.1. Setelah itu data informasi akan dikirim ke web dan jika suara yang terdeteksi adalah kondisi tenang maka *LED* hijau akan menyala, sedangkan jika yang terdeteksi adalah kondisi berisik maka *LED* kuning akan menyala, *LCD* dan modul *audio* akan aktif dan memberikan informasi pemberitahuan. Dan saat kondisi sangat berisik terdeteksi maka *LED* merah akan menyala, *LCD* dan modul *audio* akan aktif dan memberikan informasi pemberitahuan.

Tabel 3.1 Klasifikasi kondisi ruangan

Desibel (dB)	Kondisi Ruangan
0 – 50 dB	Tenang
51 – 69 dB	Berisik
≥ 70 dB	Sangat Berisik

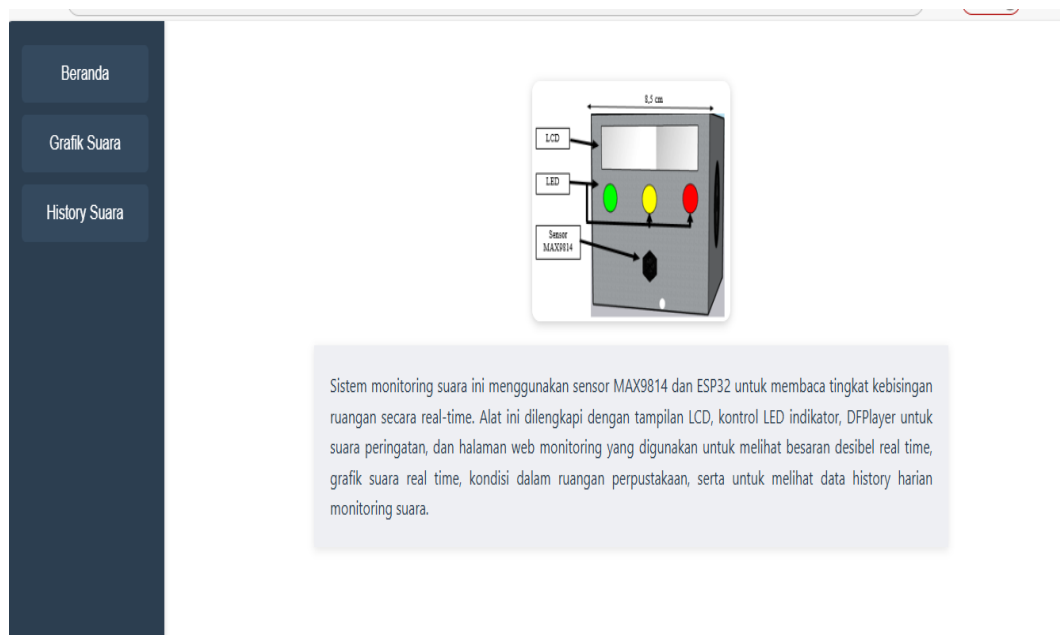
(Sumber : <https://www.pusatalatbantudengar.com/blog/tingkat-kebisingan-suara>)



Gambar 3.4 Diagram Alir Prinsip Kerja Alat

C. Tampilan Halaman Webs

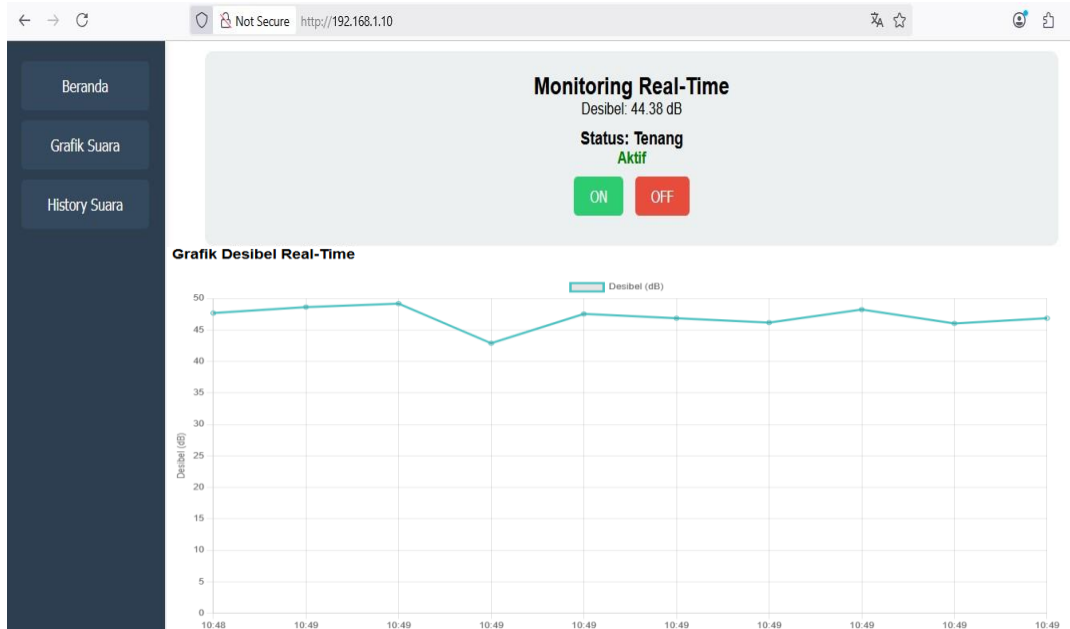
Berikut ini adalah beberapa tampilan *web* yang mencakup, halaman Beranda, halaman grafik suara, dan halaman *history* suara. Gambar 3.5 memperlihatkan tampilan halaman beranda , yang menampilkan informasi nilai *desibel* secara real time yang di deteksi oleh sensor, tombol *on off* untuk menonaktifkan alat melalui *web*, dan tampilan status alat apakah sedang aktif atau tidak.



Gambar 3. 5 Halaman Beranda

Pada gambar 3.6 dan Gambar 3.7 menampilkan tampilan pada halaman grafik suara dan *History* suara yang dimana pada menu grafik suara ditampilkan grafik suara secara *real time* yang sedang dideteksi oleh sensor, grafik ini dibuat untuk mempermudah dalam mengamati perubahan suara yang terjadi, sedangkan pada menu *history* suara menampilkan *file* suara per jam yang telah dideteksi pada hari-hari sebelumnya, *file* tersebut dikelompokkan sesuai dengan tanggal kapan *file* suara tersebut dideteksi, halaman ini dibuat untuk membantu mengamati perubahan suara

yang terjadi pada tanggal atau hari-hari sebelumnya.



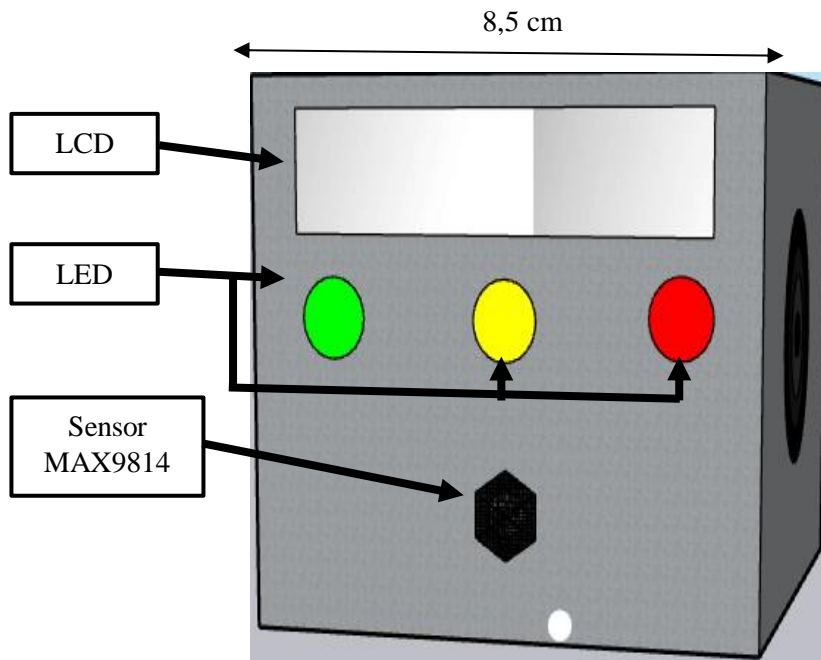
gambar 3.6 Halaman Grafik suara

Waktu	Desibel (dB)	Kondisi
12:09:33	46.58	Tenang
12:41:39	51.51	Berisik
13:00:00	44.93	Tenang
14:00:00	45.34	Tenang
15:00:00	46.71	Tenang
15:41:56	43.84	Tenang
16:00:00	41.64	Tenang
16:38:32	48.08	Tenang
17:00:00	46.03	Tenang
17:10:29	48.9	Tenang

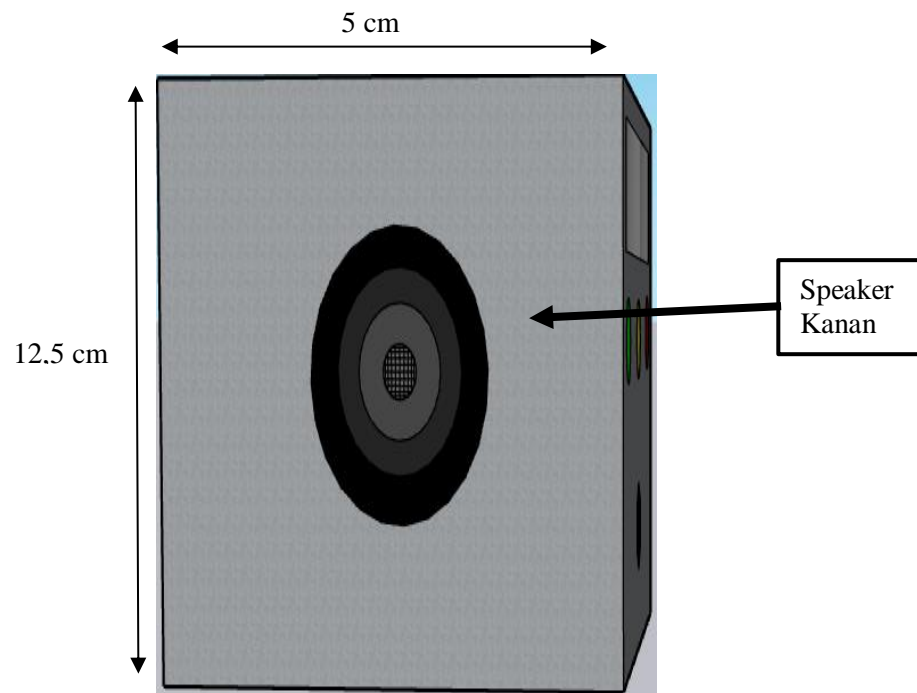
Gambar 3.7 Halaman *History* Suara

D. Desain Alat

Adapun desain alat pendeteksi kebisingan dan pemberi peringatan yang terdapat pada gambar 3.8 dan gambar 3.9



Gambar 3. 8 Rancangan desain tampak depan



Gambar 3. 9 Rancangan desain tampak samping Kanan

Gambar 3.8 dan gambar 3.9 menunjukkan bentuk rancangan desain alat pendeteksi kebisingan dan pemberi peringatan di perpustakaan berbasis web server. Pada rancangan desain ini tinggi alat yaitu 12,5 cm, panjang 5 cm, dan panjang 8,5 cm. Pada bagian depan rancangan desain alat terdapat sebuah *LCD* untuk menampilkan kondisi dalam ruangan, tiga buah *LED* sebagai indikator kondisi ruangan, dan sensor suara MAX 9814 untuk mendeteksi suara dalam ruangan. Sedangkan pada bagian samping kanan terdapat *speaker* yang digunakan untuk memberikan peringatan kepada pengunjung jika terjadi kebisingan di dalam ruangan.

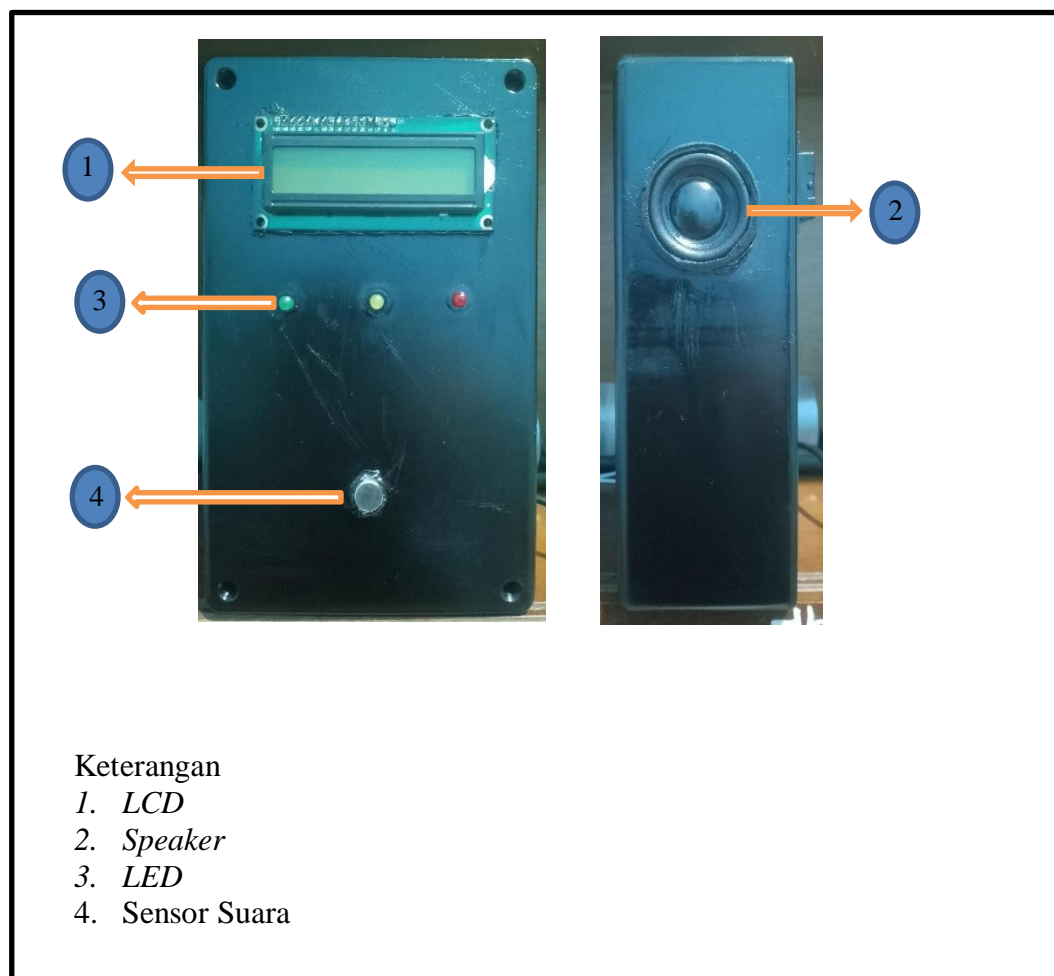
BAB IV

HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian yang berjudul “Rancang bangun alat pendeteksi kebisingan dan pemberi peringatan di perpustakaan berbasis *web server*”, maka penulis dapat menampilkan dalam bentuk fisik berupa alat, *software*, *hardware* dan data. Adapun hasil dari penelitian adalah sebagai berikut:

4.1 Bentuk Fisik Alat

Berikut ini adalah gambar bentuk fisik alat yang ditampilkan pada gambar 4.1.



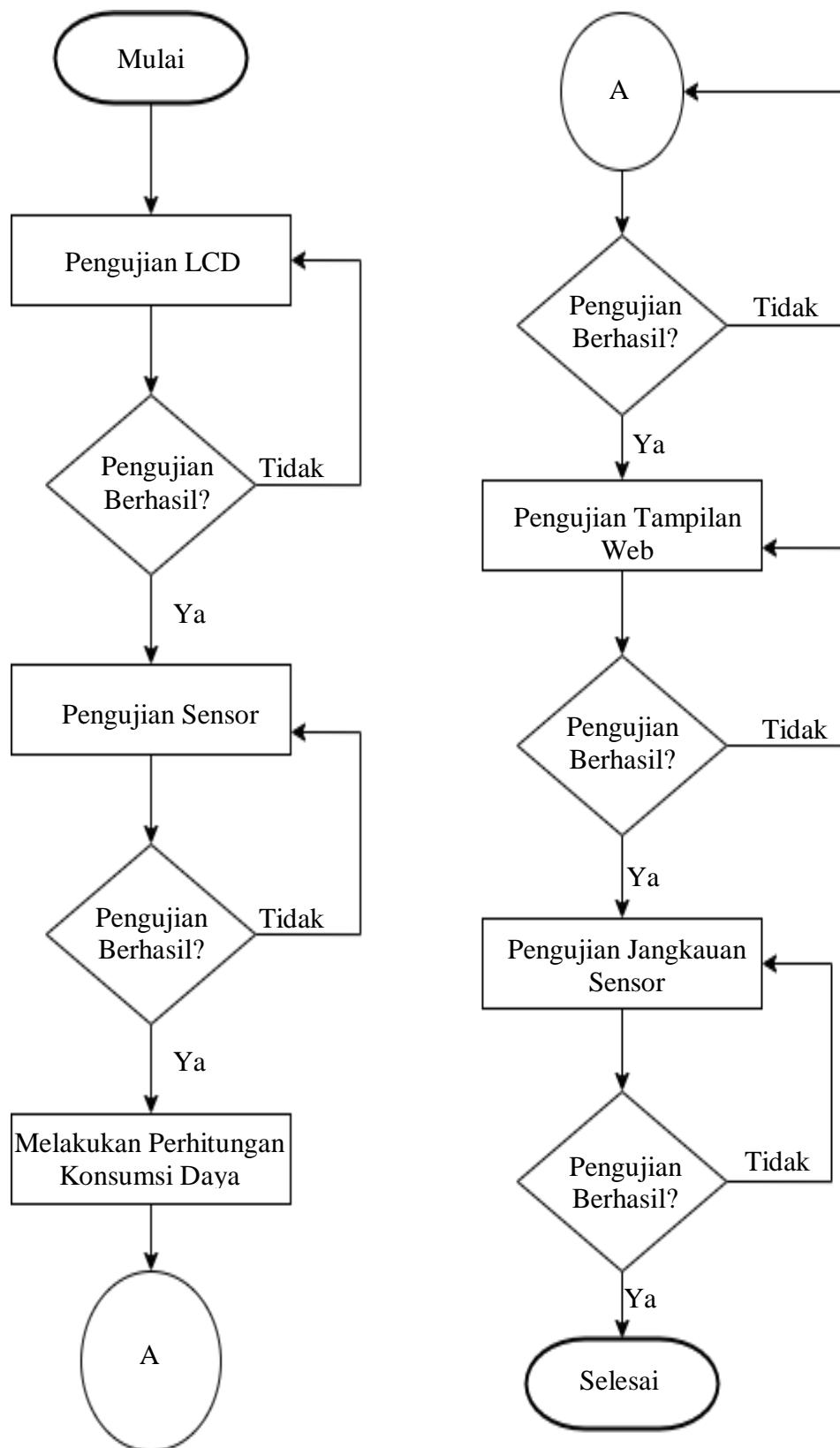
Gambar 4.1 Bentuk Fisik Alat

Gambar 4.1 menunjukkan bentuk fisik dari alat pendeteksi kebisingan dan pemberi peringatan di perpustakaan berbasis *Web Server*, Casing alat ini terbuat dari bahan mika dengan ketebalan 3 mm, yang berfungsi sebagai pelindung komponen utama.

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa perangkat keras dan perangkat lunak pada alat bekerja secara optimal sesuai dengan yang diharapkan. Pengujian mencakup berbagai percobaan untuk mengetahui apakah fungsi *input* dan *output* pada alat telah bekerja dengan benar serta mengidentifikasi kemungkinan terjadinya kesalahan atau kekurangan dalam sistem berdasarkan hasil dari setiap percobaan yang dilakukan.

Ada beberapa tahapan yang dilakukan dalam pengujian sistem, yaitu pengujian *LCD*, pengujian Sensor Suara, pengujian konsumsi daya, pengujian respon *desibel* ke *web server*, dan terakhir melakukan pengujian alat secara keseluruhan. Adapun tahapan – tahapan pengujian dari alat pendeteksi kebisingan dan pemberi peringatan di perpustakaan berbasis *web server* terdapat pada gambar 4.2 berikut ini:



Gambar 4.2 Diagram Langkah Pengujian Alat

4.2.1. Pengujian LCD

Pengujian *LCD* 16x2 dilakukan dengan tujuan untuk mendapatkan parameter berupa tampilan karakter pada *LCD* sesuai dengan keinginan dan kebutuhan yang diperlukan bagi penelitian ini. Pengujian dilakukan dengan menghubungkan rangkaian pada mikrokontroler dan memprogram karakter atau tulisan yang ingin ditampilkan pada *LCD* melalui sebuah *mikrokontroler*, dalam penelitian ini *mikrokontroler* yang digunakan adalah ESP32, Adapun tulisan yang akan ditampilkan pada layar *LCD* saat sensor mendeteksi suara adalah :

4.2.1.1 Tampilan *LCD* saat sensor mendeteksi suara <50 dB



Gambar 4.3 Tampilan *LCD* Saat Kondisi Tenang

4.2.1.2 Tampilan *LCD* saat sensor mendeteksi suara ≥ 50 dB dan ≤ 69 dB



Gambar 4.4 Tampilan *LCD* Saat Kondisi Berisik

4.2.1.3 Tampilan LCD saat sensor mendeteksi mendeteksi suara ≥ 70 dB



Gambar 4.5 Tampilan *LCD* Saat kondisi sangat berisik

Berdasarkan hasil percobaan yang telah dilakukan, sistem tampilan menggunakan layar LCD bekerja dengan baik sesuai perancangan. LCD mampu menampilkan informasi kondisi ruangan berdasarkan tingkat kebisingan yang terdeteksi sensor, yaitu kategori *tenang* (≤ 50 dB) seperti pada gambar 4.3, *berisik* (51–69 dB) seperti pada gambar 4.4, dan *sangat berisik* (≥ 70 dB) seperti pada gambar 4.5. Perubahan status pada layar LCD dapat ditampilkan secara real-time mengikuti perubahan suara di lingkungan perpustakaan, sehingga pengguna dapat langsung mengetahui kondisi kebisingan ruangan pada saat itu.

4.2.2. Pengujian Sensor Suara

Sensor suara MAX9814 bekerja apabila adanya suara atau sumber suara, untuk menentukan kepekaan dan keakuratan sensor maka dilakukan kalibrasi dengan membandingkan nilai analog sensor dengan nilai dB pada alat ukur. Alat ukur yang digunakan untuk membandingkan nilai analog pada sensor dengan nilai dB pada alat ukur adalah *Sound level meter*. Setelah nilai pada sensor dan *Sound Level Meter* didapatkan maka akan dihitung *error* dan rata-rata dengan menggunakan rumus

error :

$$Error = \left| \frac{\text{Nilai sensor} - \text{Nilai alat ukur}}{\text{Nilai sensor}} \right| \times 100\% \dots\dots\dots(4.1)$$

$$\text{Rata - rata } Error = \left(\frac{\text{Jumlah Nilai Error}}{\text{Banyaknya Pengujian}} \right) \dots\dots\dots(4.2)$$

Tabel 4.1 Validasi sensor suara

No	Sound Level Meter (dB)	Sensor MAX9814 (dB)	Error (%)
1	30	31,2	3,85
2	35	36,3	3,58
3	40	42,1	4,99
4	45	46,4	3,02
5	50	51,1	2,15
6	55	56,7	2,99
7	60	60,8	1,32
8	65	66,4	2,11
9	70	71,3	1,82
	Rata-rata error		2.87%

Tabel diatas menunjukkan hasil dari pengujian kebisingan yang telah dilakukan, seperti pada tabel dilakukan pengujian sebanyak pada rentang *desibel* 30 dB – 70 dB dan didapatkan rata-rata *error* sensor suara MAX9814 yaitu berkisar 2,87% yang menandakan sensor suara cukup akurat dalam mendeteksi suara.

4.2.3. Pengujian Konsumsi Daya

Pada pengujian ini, dilakukan pengujian konsumsi daya agar dapat mengetahui total daya komponen yang akan digunakan. Berikut rumus yang digunakan untuk menghitung daya yang digunakan komponen:

$$W = V \times I \dots \dots \dots (4.3)$$

Keterangan :

W = Daya (Watt) V = Tegangan (Volt) I = Arus (Ampere)

Tabel 4.2 Hasil Perhitungan Konsumsi Daya

No	Nama komponen	Volt (V)	Arus (A)	Daya (W)
1	ESP32	5	0,5	2,5
2	Sensor MAX 9814	2,7	0,005	0,013
3	LCD 16x2	5	0,003	1,5
4	DFPlayer	5	0,3	0,13
5	LED Kuning	2,2	0,02	0,04
6	LED Merah	2,2	0,02	0,04
7	LED Hijau	3,2	0,02	0,06
	Total Daya			4,3

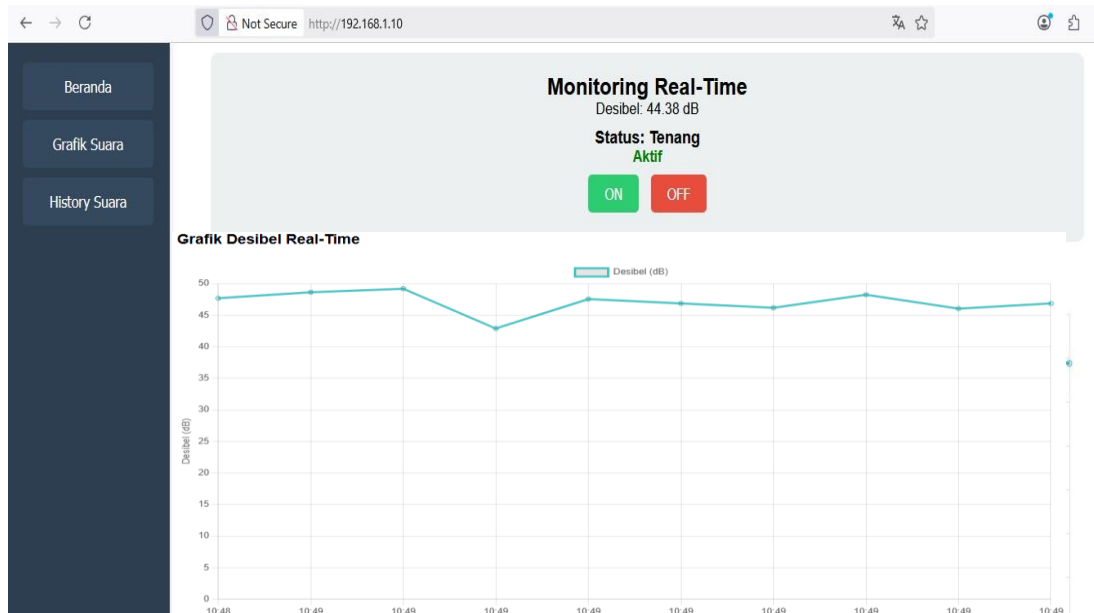
Tabel 4.2 menunjukkan konsumsi daya pada masing – masing komponen alat. Pada komponen ESP32 berfungsi sebagai modul *mikrokontroler* utama yang mengendalikan seluruh sistem, termasuk pemrosesan data dari sensor dan komunikasi dengan *server* menggunakan daya 2,5 watt. Sensor MAX9814 berfungsi sebagai alat untuk mendeteksi perubahan suara yang terjadi di ruangan perpustakaan. Konsumsi daya sensor ini cukup rendah, yaitu 0,013 watt. LCD 16x2 digunakan sebagai tampilan kondisi dalam ruangan, seperti tenang, berisik, dan sangat berisik. Dengan konsumsi daya 0,015 watt. DFPlayer Merupakan modul pemutar audio yang berfungsi sebagai pemberi peringatan ketika terdeteksi kebisingan di ruang perpustakaan yang melebihi batas *desibel* yang ditentukan, Komponen ini memiliki konsumsi daya sebesar 1,5 watt. LED kuning digunakan

sebagai indikator bahwa ruangan sedang dalam kondisi berisik, *LED* warna kuning ini memiliki konsumsi daya sebesar 0,04 *watt*. *LED* merah digunakan sebagai indikator bahwa ruangan sedang dalam kondisi sangat berisik. *LED* warna merah ini memiliki jumlah konsumsi daya sebesar 0,04 *watt*. *LED* hijau digunakan sebagai indikator bahwa ruangan sedang dalam kondisi tenang atau *desibel* yang dideteksi < 50 *dB*. *LED* ini memiliki jumlah konsumsi daya sebesar 0,06 *watt*. Dari perhitungan daya yang dilakukan didapatkan total konsumsi daya sebesar 4,3 *watt*

4.2.4. Pengujian Tampilan Web dan Real-Time Grafik

Pengujian ini bertujuan untuk memastikan bahwa sistem web server lokal yang dibangun pada ESP32 dapat menampilkan data suara secara *real-time*, menampilkan grafik perubahan desibel, serta menyajikan data riwayat pengukuran yang telah tersimpan.

Langkah awal pengujian dilakukan dengan mengakses alamat IP ESP32 melalui browser pada perangkat yang terhubung ke jaringan *WiFi* yang sama. Setelah berhasil mengakses halaman utama, kemudian memilih menu “Grafik Suara” untuk melihat grafik pergerakan nilai *desibel* secara *real-time*. Selama pengujian, berbagai kondisi suara diberikan, seperti kondisi hening, percakapan, dan suara keras. Perubahan suara tersebut dapat langsung terlihat pada grafik di halaman *web* seperti pada gambar 4.6



Gambar 4.6 Menu Grafik Suara

Nilai *desibel* yang ditampilkan juga telah dibandingkan dengan nilai pada *Serial Monitor*, dan hasilnya menunjukkan konsistensi data antara tampilan web dan sistem *monitoring* fisik.

Pengujian juga dilakukan pada menu “*History Suara*”. Ketika penggunaan memilih *file* data harian (misalnya, 2025-05-25.txt), table *desibel* harian ditampilkan berdasarkan data yang tersimpan di SPIFFS seperti pada gambar 4.7

Waktu	Desibel (dB)	Kondisi
12:09:33	46.58	Tenang
12:41:39	51.51	Berisik
13:00:00	44.93	Tenang
14:00:00	45.34	Tenang
15:00:00	46.71	Tenang
15:41:56	43.84	Tenang
16:00:00	41.64	Tenang
16:38:32	48.08	Tenang
17:00:00	46.03	Tenang
17:10:29	48.9	Tenang

Gambar 4.7 Menu *History Suara* Harian

Secara keseluruhan, hasil pengujian menunjukkan bahwa *web server* pada ESP32 dapat diakses dengan lancar, grafik *real-time* dapat menampilkan perubahan *desibel* secara akurat, dan data riwayat dari SPIFFS dapat ditampilkan dengan baik melalui antarmuka *web*, Hasil pengujian dapat dilihat pada tabel 4.3

Tabel 4.3 Pengujian Web

No	Halaman	Aksi	Hasil Tampilan	Grafik Muncul	Data Akurat	Tabel Muncul	Catatan
1	Beranda	Akses web pertama	Status sistem tampil	-	Ya	-	Web server aktif
2	Grafik Suara	Suara berubah (diam → keras)	Grafik bergerak	Ya	Ya	-	Data sesuai dengan Serial Monitor
3	History Suara	Klik tanggal 2025-05-25.txt	Grafik jam 08–17 tampil	-	Ya	Ya	Data valid

4.2.5. Pengujian Jangkauan Sensor Suara

Pengujian ini bertujuan untuk mengetahui sejauh mana sensor mampu mendeteksi tingkat intensitas suara (dB) pada berbagai jarak terhadap sumber suara. Dengan mengetahui karakteristik jangkauan sensor, maka dapat ditentukan tingkat keandalan alat, khususnya pada ruangan dengan variasi sumber kebisingan seperti perpustakaan. Untuk memastikan konsistensi percobaan, digunakan sumber suara berupa speaker handphone dengan volume konstan sebagai pembangkit bunyi. Sensor ditempatkan pada posisi sejajar dengan speaker, kemudian dilakukan pengukuran secara bertahap mulai dari jarak 1 meter hingga 5 meter. Pemilihan jarak ini didasarkan pada pertimbangan bahwa dalam penggunaan nyata, sumber

kebisingan dapat berada pada rentang dekat hingga menengah dari sensor. Nilai intensitas suara pada jarak 1 meter ditetapkan sebagai acuan awal, sedangkan nilai pada jarak berikutnya akan dianalisis berdasarkan penurunan level suara. Data percobaan bisa dilihat pada tabel 4.4.

Tabel 4.4 Jarak Deteksi sensor

Sumber Suara	Jarak (m)	Percobaan Pertama (dB)	Percobaan Kedua (dB)	Percobaan Ketiga (dB)	Percobaan Keempat (dB)
Speaker handphone (volume konstan)	1	50.0	55.0	69.0	70.0
Speaker handphone (volume konstan)	2	44.1	49.2	63.1	64.1
Speaker handphone (volume konstan)	3	40.3	45.5	59.7	60.5
Speaker handphone (volume konstan)	4	38.1	43.4	57.4	58.3
Speaker handphone (volume konstan)	5	36.2	41.2	55.3	56.7

Berdasarkan hasil pengukuran, tingkat kebisingan yang dideteksi sensor MAX9814 mengalami penurunan seiring dengan bertambahnya jarak antara sumber suara dan sensor. Pada jarak 1 meter, nilai awal yang terukur ditetapkan sebesar 50 dB, 55 dB, 69 dB, dan 70 dB. Ketika jarak diperbesar menjadi 2 meter, tingkat desibel menurun sekitar 6 dB dari nilai awal. Penurunan yang sama terlihat pada jarak 3 meter hingga 5 meter, meskipun semakin jauh, selisih penurunan menjadi lebih kecil (tidak linear, melainkan mengikuti hukum inverse-square).

Hal ini menunjukkan bahwa sensor MAX9814 dapat mendeteksi perubahan intensitas suara dengan cukup baik. Pada skenario dengan nilai awal 50 dB, hasil pengukuran pada jarak 5 meter hanya sekitar 36,2 dB, sementara pada nilai awal 70 dB di jarak yang sama masih terbaca sekitar 56,7 dB. Artinya, semakin besar

intensitas awal suara, semakin tinggi pula tingkat desibel yang masih dapat dideteksi sensor pada jarak jauh.

4.2.6. Pengujian Alat Keseluruhan

Pengujian serta Pengambilan data dalam penelitian ini dilakukan selama lima hari di area perpustakaan Universitas Tadulako, pengujian dilakukan bertujuan untuk memastikan bahwa alat dan sistem yang dirancang dapat bekerja sesuai dengan yang diinginkan. Data yang dikumpulkan berupa tingkat kebisingan suara yang diukur dalam satuan *desibel* (dB), dengan interval pencatatan setiap satu jam, dimulai dari pukul 12.00 hingga 16.00 WITA, kemudian akan dihitung rata-rata desibel dalam tiap harinya menggunakan rumus

$$\text{Rata - rata Desibel} = \left(\frac{\text{Jumlah Desibel Tiap Jam}}{\text{Banyaknya Data Desibel}} \right) \dots\dots\dots(4.4)$$

Proses pengambilan data dilakukan secara otomatis menggunakan sistem monitoring berbasis *mikrokontroler* yang telah dirancang, data yang didapatkan dapat dilihat pada tabel 4.4

Tabel 4.5 Monitoring Suara Harian

Hari	Jam	Desibel (dB)	Kondisi Ruangan
Pertama	12.00	36,95	Tenang
	13.00	36,34	Tenang
	14.00	23,66	Tenang
	15.00	39,31	Tenang
	16.00	35,74	Tenang
Rata-rata		34,4	Tenang
Kedua	12.00	34,75	Tenang
	13.00	36,93	Tenang
	14.00	33,96	Tenang

	15.00	32,38	Tenang
	16.00	42,87	Tenang
Rata-rata		36,18	Tenang
Ketiga	12.00	27,23	Tenang
	13.00	32,18	Tenang
	14.00	33,17	Tenang
	15.00	36,73	Tenang
	16.00	37,72	Tenang
Rata-rata		33,4	Tenang
Keempat	12.00	31,50	Tenang
	13.00	34,20	Tenang
	14.00	30,80	Tenang
	15.00	35,60	Tenang
	16.00	38,10	Tenang
Rata-rata		34,84	Tenang
Kelima	12.00	29,70	Tenang
	13.00	33,40	Tenang
	14.00	32,10	Tenang
	15.00	37,50	Tenang
	16.00	36,20	Tenang
Rata-rata		33,78	Tenang

Hasil monitoring pada table 4.4 menunjukkan hasil pengukuran tingkat kebisingan di gedung perpustakaan Universitas Tadulako yang dilakukan selama lima hari berturut-turut pada rentang waktu pukul 12.00 hingga 16.00, diperoleh rata-rata tingkat kebisingan harian sebagai berikut: hari pertama sebesar 34,4 dB, hari kedua sebesar 36,18 dB, hari ketiga sebesar 33,4 dB, hari keempat sebesar 34,84 dB, dan hari kelima sebesar 33,78 dB.

Apabila dibandingkan dengan klasifikasi kondisi ruangan berdasarkan tingkat kebisingan, seluruh data yang diperoleh berada pada kategori tenang (≤ 50 dB). Rentang nilai yang terukur yaitu antara 23,66 dB hingga 42,87 dB, menunjukkan bahwa aktivitas di dalam perpustakaan relatif stabil dan tidak menimbulkan gangguan kebisingan.

Dengan demikian, dapat disimpulkan bahwa kondisi lingkungan perpustakaan Universitas Tadulako pada waktu pengukuran tergolong kondusif untuk kegiatan membaca, belajar, dan aktivitas akademik lainnya, karena tingkat kebisingan masih berada di bawah ambang batas gangguan konsentrasi yaitu 50 dB.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dengan judul sistem Rancang Bangun Alat Pendeteksi Kebisingan Dan Pemberi Peringatan Di Perpustakaan Berbasis Web Server, maka dapat diambil kesimpulan sebagai berikut:

1. Pengujian alat secara keseluruhan sistem pendeteksi kebisingan dan pemberi peringatan berbasis *web server* sudah bekerja dengan baik, sistem berhasil mendeteksi besaran *desibel* dalam ruangan perpustakaan secara *real time* dan mengirimkannya ke web sehingga dapat dipantau dilihat secara langsung.
2. *Website* yang dibuat bekerja dengan baik dalam menampilkan *desibel* suara secara *real time*, menampilkan status alat, menampilkan grafik *desibel* suara secara *real time*, dan dapat menampilkan *history* suara yang sudah tersimpan sebelumnya.
3. Pembacaan *desibel* suara antara sensor dan alat *Sound Level Meter* memiliki rata-rata *error* yaitu sebesar 2,87%.
4. Kondisi pada ruangan perpustakaan Universitas Tadulako masih tergolong kondusif, karena masih memiliki nilai tingkat kebisingan dibawah 50 dB

5.2 Saran

Berdasarkan hasil perancangan dan penelitian yang telah penulis lakukan pada tugas akhir ini maka dapat diberikan saran sebagai berikut:

1. Untuk pembacaan suara agar lebih akurat, sebaiknya letakkan sensor di tiap-

tiap meja yang ada didalam perpustakaan 1 meja 1 sensor suara.

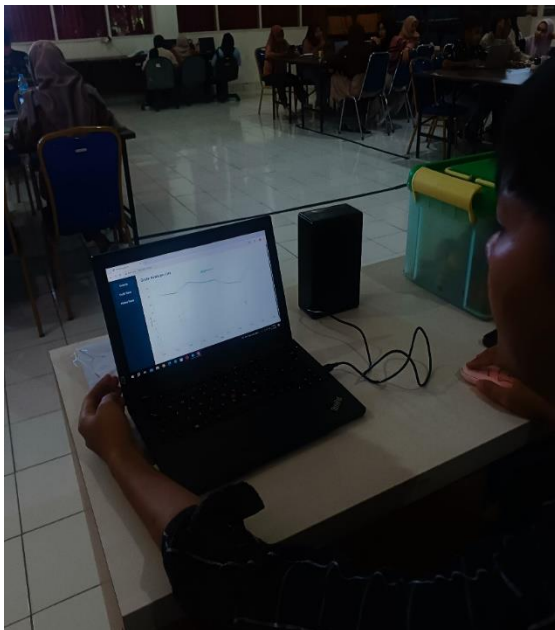
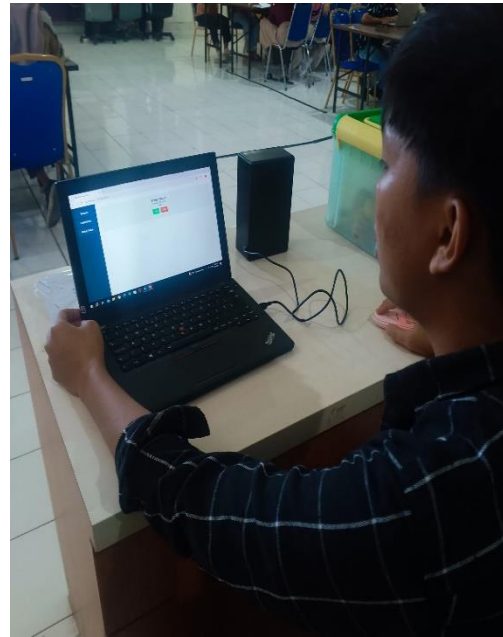
2. Sistem dapat dilengkapi dengan notifikasi *real time* melalui *whatsapp* atau *email* jika terjadi kebisingan yang melebihi ambang batas dalam ruangan.
3. Penggunaan sensor suara yang dengan sensitivitas yang lebih tinggi dan pengolahan sinyal digital yang lebih kompleks juga disarankan untuk meningkatkan akurasi deteksi desibel dan stabilitas sistem.

DAFTAR PUSTAKA

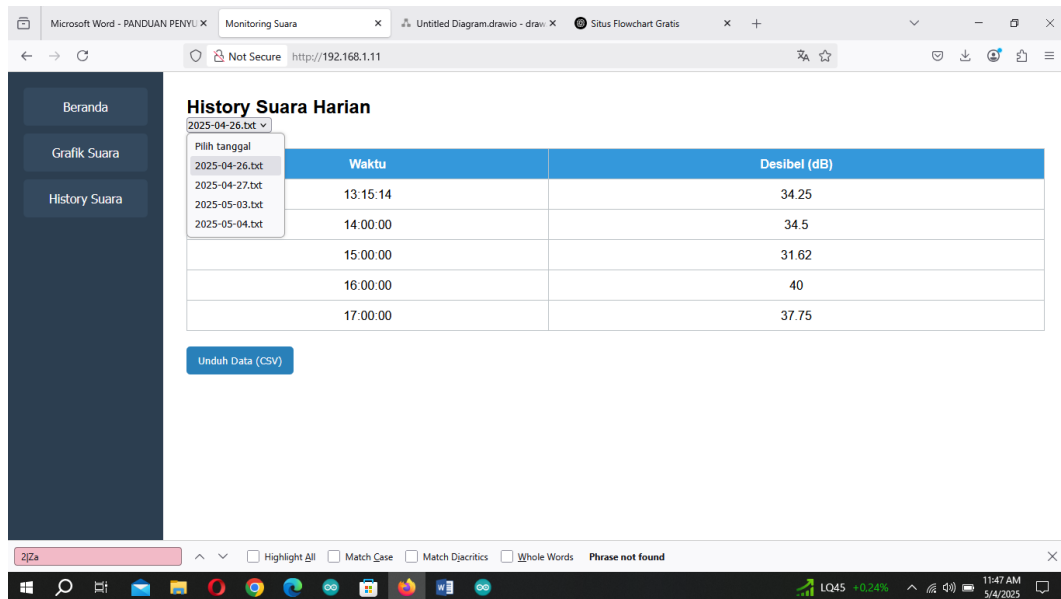
- Ardutech. 2020. Mengenal ESP32 Development Kit untuk IoT (*Internet of Things*), <https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>, diakses 6 September 2024
- Bahri, K. (2017). *Dasar Pemrograman C/C++*. Levatra.Com. <https://www.levatra.com/2017/03/konsep-dasar-bahasa-pemrograman-c.html>, diakses 18 Agustus 2024
- Bersama, E. (2024). *LED Atau light Emitting Diode*. Elektronika Bersama.Com. <https://www.elektronikabersama.web.id/2011/05/led-atau-light-emitting-diode.html>, diakses 18 Agustus 2024
- Dispersip. (2018). *Perpustakaan Sekolah Yang Terlupakan*. Dispersip Kabupaten Kampar. <https://pustakaarsip.kamparkab.go.id/artikeldetail/1379/perpustakaan-sekolah-yang-terlupakan>, diakses 18 Agustus 2024
- Ikhwani, M., Harmadi, H., & Irfandy, D. (2021). Rancang Bangun Alat pendeteksi Tingkat Tekanan Bunyi Terhadap Respon Membran Timpani Berbasis Serat Optik. *Jurnal Fisika Unand*, 10(3), 274–280. <https://doi.org/10.25077/jfu.10.3.274-280.2021>
- Keliat, Windarwati, pawiroyono dan subu 2015. (2015). Pemenuhan Kebutuhan Rasa Nyaman. *Repository.Poltekkes-Denpasar.Ac.Id*, 53(9), 1689–1699. [http://repository.poltekkes-denpasar.ac.id/2284/3/BAB II.pdf](http://repository.poltekkes-denpasar.ac.id/2284/3/BAB%20II.pdf)
- Munawir, A., Murhaban, M., & Zulfan, Z. (2020). Pengukuran Sound Pressure Level (SPL) dalam kabin mobil dengan menggunakan SEA Model. *Jurnal Mekanova: Mekanikal, Inovasi Dan Teknologi*, 6(2), 117. <https://doi.org/10.35308/jmkn.v6i2.2758>
- Suardana, I., Swardika, I. K., & Wiranata, L. F. (2022). Sistem Deteksi Pelanggaran Pengendara dengan Computer Vision Sesi Posisi Kendaraan dan Batas Kebisingan Kendaraan. *Repository Politeknik Negeri Bali, September*, 1–8. http://repository.pnb.ac.id/id/eprint/3235%0Ahttp://repository.pnb.ac.id/3235/1/RAMA_36304_1815344022_artikel.pdf
- Syihabuddin, A., Minto Basuki, B., & Sugiono. (2022). *Alat Pendeteksi Kebisingan Di Ruang Pasien Berbasis Fuzzy Logic*.
- Widyadana, K. F. A. (2021). Perancangan Sistem Kebisingan Suara Dengan Sound Sensor Mic Berbasis Arduino. *Researchgate.Net*, April. <https://www.researchgate.net/profile/Raditya-Pratama-6/publication/>

LAMPIRAN

1. Dokumentasi Pengambilan Data Di Gedung Perpustakaan Universitas Tadulako



2. Data Hasil



3. Program

3.1 Program Alat

```
#include <WiFi.h>
#include <WebServer.h>
#include <LiquidCrystal_I2C.h>
#include <SPIFFS.h>
#include <time.h>
#include <DFRobotDFPlayerMini.h>
#include <HardwareSerial.h>

// WiFi
const char* ssid = "Alat";
const char* password = "123456789";

// Komponen
#define MAX9814_PIN 32
#define LED_HIJAU 4
#define LED_KUNING 2
#define LED_MERAH 17
LiquidCrystal_I2C lcd(0x27, 16, 2);
HardwareSerial dfSerial(1); // Gunakan UART1
DFRobotDFPlayerMini dfplayer;

// Batas
#define THRESHOLD_HIJAU_MAX 1349
#define THRESHOLD_KUNING_MIN 1350
#define THRESHOLD_KUNING_MAX 1399
```

```

#define THRESHOLD_MERAH 1400

// Variabel
int analogValue = 0;
float desibel = 0;
bool wasNoisy = false;
unsigned long lastWarningTime = 0;
unsigned long lastUpdateTime = 0;
unsigned long updateInterval = 1000;
int lastSavedHour = -1;
bool alatAktif = true; // untuk kontrol ON/OFF
String lastStatus = "";

// Web Server
WebServer server(80);

// Sinkronisasi waktu NTP
void setupTime() {
    configTime(7 * 3600, 0, "pool.ntp.org", "time.nist.gov");
    struct tm timeinfo;
    if (getLocalTime(&timeinfo)) {
        Serial.println("Waktu tersinkron");
    } else {
        Serial.println("Gagal sinkron waktu");
    }
}

// Simpan data ke SPIFFS per jam
void simpanData(float desibel) {
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        Serial.println("[GAGAL] Tidak bisa ambil waktu");
        return;
    }

    if (timeinfo.tm_hour < 8 || timeinfo.tm_hour > 17) {
        Serial.println("[INFO] Di luar jam 08:00 - 17:00, data tidak disimpan");
        return;
    }

    if (timeinfo.tm_hour == lastSavedHour) {
        return; // Sudah disimpan jam ini
    }

    char waktu[9];
    strftime(waktu, sizeof(waktu), "%H:%M:%S", &timeinfo);
    file.printf("%s,%.2f\n", waktu, desibel);
    file.close();

    Serial.printf("[SUKSES] Data disimpan: %s = %.2f dB\n", waktu, desibel);
}

```

```

    lastSavedHour = timeinfo.tm_hour;
}

// TASK 1: Sensor
void TaskSensor(void *pvParameters) {

    Serial.print("[DEBUG] Sisa stack TaskSensor: ");
    Serial.println(uxTaskGetStackHighWaterMark(NULL));

    while (true) {
        if (alatAktif) {
            long sum = 0;
            for (int i = 0; i < 10; i++) {
                sum += analogRead(MAX9814_PIN);
                vTaskDelay(pdMS_TO_TICKS(5));
            }
            analogValue = sum / 10;

            float slope = (70.0 - 50.0) / (1400 - 1349);
            float intercept = 50.0 - (slope * 1349);
            desibel = slope * analogValue + intercept;

            if (millis() - lastUpdateTime >= updateInterval) {
                Serial.print("Analog: ");
                Serial.print(analogValue);
                Serial.print(" | Desibel: ");
                Serial.println(desibel);
                lastUpdateTime = millis();
            }

            simpanData(desibel);
        }
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
}

// TASK 2: Output
void TaskOutput(void *pvParameters) {
    while (true) {
        if (alatAktif) {
            String currentStatus = "";

            if (desibel <= 50) {
                digitalWrite(LED_HIJAU, HIGH);
                digitalWrite(LED_KUNING, LOW);
                digitalWrite(LED_MERAH, LOW);
                currentStatus = "Tenang";
                wasNoisy = false;
            } else if (desibel >= 51 && desibel <= 69) {
                digitalWrite(LED_HIJAU, LOW);

```

```

digitalWrite(LED_KUNING, HIGH);
digitalWrite(LED_MERAH, LOW);
currentStatus = "Berisik";
if (!wasNoisy) {

    dfplayer.play(1);
    wasNoisy = true;
    lastWarningTime = millis();
}
if (millis() - lastWarningTime > 5000) {
    wasNoisy = false;
}
} else if (desibel >= 70) {
digitalWrite(LED_HIJAU, LOW);
digitalWrite(LED_KUNING, LOW);
digitalWrite(LED_MERAH, HIGH);
currentStatus = "Sangat Berisik";

if (!wasNoisy) {
    dfplayer.play(1);
    wasNoisy = true;
    lastWarningTime = millis();
}

if (millis() - lastWarningTime > 5000) {
    wasNoisy = false;
}
}

// Update LCD hanya jika status berubah
if (currentStatus != lastStatus) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(currentStatus);
    lastStatus = currentStatus;
}
lcd.setCursor(0, 1);
lcd.print(WiFi.localIP().toString().c_str());
}
vTaskDelay(pdMS_TO_TICKS(200));
}

}

// TASK 3: Web
void TaskWeb(void *pvParameters) {
    server.on("/", HTTP_GET, []() {
        File file = SPIFFS.open("/index.html", "r");
        server.streamFile(file, "text/html");
        file.close();
    });
}

```

```

server.on("/style.css", HTTP_GET, []() {
    File file = SPIFFS.open("/style.css", "r");
    server.streamFile(file, "text/css");
    file.close();
});
server.on("/script.js", HTTP_GET, []() {
    File file = SPIFFS.open("/script.js", "r");
    server.streamFile(file, "application/javascript");
    file.close();
});
server.on("/data", HTTP_GET, []() {
    String json = "{\"analog\":\"" + String(analogValue) + "\",\"desibel\":\"" + String(desibel) +
    "\"}";
    server.send(200, "application/json", json);
});
server.on("/list", HTTP_GET, []() {
    String json = "[";
    File root = SPIFFS.open("/");
    File file = root.openNextFile();
    bool first = true;
    while (file) {
        String name = file.name();
        if (name.endsWith(".txt")) {
            if (!first) json += ",";
            json += "\"" + name + "\"";
            first = false;
        }
        file = root.openNextFile();
    }
    json += "]";
    server.send(200, "application/json", json);
});
server.on("/grafik", HTTP_GET, []() {
    if (!server.hasArg("tanggal")) {
        server.send(400, "text/plain", "Parameter tanggal diperlukan");
        return;
    }

    String tanggal = "/" + server.arg("tanggal");
    File file = SPIFFS.open(tanggal, "r");

    if (!file) {
        server.send(404, "text/plain", "File tidak ditemukan");
        return;
    }

    String json = "[";
    bool first = true;
    while (file.available()) {
        String line = file.readStringUntil('\n');

```

```

    line.trim();
    if (line.length() == 0) continue;
    int idx = line.indexOf(',');
    if (idx != -1) {
        String waktu = line.substring(0, idx);
        String db = line.substring(idx + 1);
        if (!first) json += ",";
        json += "{\"waktu\": \"" + waktu + "\", \"desibel\": " + db + "}";
        first = false;
    }
}
json += "]";
file.close();

server.send(200, "application/json", json);
});

// Endpoint untuk realtime grafik
server.on("/realtime", HTTP_GET, []() {
    struct tm timeinfo;
    getLocalTime(&timeinfo);
    char waktu[6];
    strftime(waktu, sizeof(waktu), "%H:%M", &timeinfo);

    String json = "{\"desibel\": " + String(desibel, 2) + ", \"waktu\": \"" + String(waktu) +
    "\"}";
    server.send(200, "application/json", json);
});

// Endpoint kontrol alat
server.on("/status", HTTP_GET, []() {
    if (server.hasArg("set")) {
        String set = server.arg("set");
        if (set == "on") alatAktif = true;
        else if (set == "off") alatAktif = false;
    }
    String json = "{\"status\": \"" + String(alatAktif ? "on" : "off") + "\"}";
    server.send(200, "application/json", json);
});

server.begin();
while (true) {
    server.handleClient();
    vTaskDelay(pdMS_TO_TICKS(10));
}
}

// Setup
void setup() {
    Serial.begin(115200);

```

```

pinMode(LED_HIJAU, OUTPUT);
pinMode(LED_KUNING, OUTPUT);
pinMode(LED_MERAH, OUTPUT);
lcd.init();
lcd.backlight();

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi");

Serial.print("ESP32 IP Address: ");
Serial.println(WiFi.localIP());

SPIFFS.begin();
setupTime();

xTaskCreate(TaskSensor, "TaskSensor", 4096, NULL, 1, NULL);
xTaskCreate(TaskOutput, "TaskOutput", 2048, NULL, 1, NULL);
xTaskCreate(TaskWeb, "TaskWeb", 8192, NULL, 1, NULL);

dfSerial.begin(9600, SERIAL_8N1, 26, 27); // RX=26, TX=27 (ubah sesuai wiring Anda)
if (!dfplayer.begin(dfSerial)) {
  Serial.println("Gagal menginisialisasi DFPlayer Mini");
} else {
  dfplayer.volume(27); // Atur volume (0-30)
  Serial.println("DFPlayer Mini siap");
}
}

// Loop
void loop() {
  // Tidak digunakan
}

```

3.2 Program Web

3.2.1 HTML

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Monitoring Suara</title>
  <link rel="stylesheet" href="style.css" />
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</style>
  /* Tambahan untuk class hidden */

```

```

.hidden {
  display: none;
}
</style>
</head>
<body>
  <div class="container">
    <!-- Sidebar -->
    <nav class="sidebar">
      <button onclick="showMenu('beranda')">Beranda</button>
      <button onclick="showMenu('grafik')">Grafik Suara</button>
      <button onclick="showMenu('history')">History Suara</button>
    </nav>
    <!-- Konten Utama -->
    <main class="content">

      <!-- Beranda -->
      <div id="beranda" class="menu">
        <div class="status-container">
          <h2>Status Suara</h2>
          <p>Desibel: <span id="desibelNow">- dB</span></p>
          <p id="statusText" class="status-tidak-aktif">Tidak Aktif</p>
          <button class="button-on" onclick="setDeviceState('on')">ON</button>
          <button class="button-off" onclick="setDeviceState('off')">OFF</button>
        </div>
      </div>

      <!-- Grafik Suara -->
      <div id="grafik" class="menu hidden">
        <h2>Grafik Suara per Jam</h2>
        <canvas id="soundChart" width="400" height="200"></canvas>
      </div>

      <!-- History -->
      <div id="history" class="menu hidden">
        <h2>History Suara Harian</h2>
        <select id="historyFileSelect" onchange="loadSelectedHistoryFile()">
          <option value="">Pilih tanggal</option>
        </select>
        <table>
          <thead>
            <tr>
              <th>Waktu</th>
              <th>Desibel (dB)</th>
            </tr>
          </thead>
          <tbody id="historyTableBody">
            <!-- Data tabel akan dimasukkan di sini oleh JavaScript -->
          </tbody>
        </table>
      </div>
    </main>
  </div>
</body>
</html>

```



```

    <p id="noDataMessage" style="text-align:center; color: gray; display: none;">Tidak
    ada data untuk tanggal ini.</p>
    <button class="export-btn" onclick="exportToCSV()">Unduh Data (CSV)</button>
  </div>
</main>
</div>
<script>
function showMenu(menu) {
  document.querySelectorAll('.menu').forEach(m => {
    m.classList.add('hidden'); // Sembunyikan semua menu
  });
  document.getElementById(menu).classList.remove('hidden'); // Tampilkan menu
yang dipilih
}

function setDeviceState(state) {
  fetch(`/status?set=${state}`)
    .then(res => res.json())
    .then(data => {
      const statusText = document.getElementById("statusText");
      if (data.status === "on") {
        statusText.textContent = "Aktif";
        statusText.className = "status-aktif";
      } else {
        statusText.textContent = "Tidak Aktif";
        statusText.className = "status-tidak-aktif";
      }
    })
    .catch(err => console.error('Gagal set status:', err));
}

function fetchRealtimeData() {
  fetch("/realtime")
    .then(res => res.json())
    .then(data => {
      const desibel = typeof data.desibel === 'object' ? data.desibel.parsedValue :
data.desibel;
      document.getElementById("desibelNow").textContent = `${desibel} dB`;

      if (typeof chart !== "undefined" && chart) {
        chart.data.labels.push(data.waktu);
        chart.data.datasets[0].data.push(desibel);
        if (chart.data.labels.length > 10) {
          chart.data.labels.shift();
          chart.data.datasets[0].data.shift();
        }
        chart.update();
      }
    })
    .catch(err => console.error('Gagal fetch data realtime:', err));
}

```

```

}

let chart;
document.addEventListener("DOMContentLoaded", () => {
  showMenu('beranda'); // Default buka Beranda

  const ctx = document.getElementById('soundChart').getContext('2d');
  chart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: [],
      datasets: [{
        label: 'Desibel (dB)',
        borderColor: 'rgb(75, 192, 192)',
        data: [],
        fill: false
      }]
    },
    options: {
      responsive: true,
      scales: {
        x: {
          title: {
            display: true,
            text: 'Waktu'
          }
        },
        y: {
          beginAtZero: true,
          title: {
            display: true,
            text: 'Desibel (dB)'
          }
        }
      }
    }
  });

  setInterval(fetchRealtimeData, 3000);
  loadHistoryFiles();
});

function loadHistoryFiles() {
  fetch("/list")
    .then(response => response.json())
    .then(files => {
      const select = document.getElementById("historyFileSelect");
      select.innerHTML = '<option value="">Pilih tanggal</option>';
      files.forEach(file => {
        const opt = document.createElement("option");
        opt.value = file;

```

```

        opt.textContent = file;
        select.appendChild(opt);
    });
})
.catch(err => console.error('Gagal load list file:', err));
}

function loadSelectedHistoryFile() {
    const fileName = document.getElementById("historyFileSelect").value;
    const tableBody = document.getElementById("historyTableBody");
    const noDataMessage = document.getElementById("noDataMessage");

    if (!fileName) return;

    fetch(`/grafik?tanggal=${fileName}`)
        .then(res => res.json())
        .then(data => {
            tableBody.innerHTML = "";
            if (data.length === 0) {
                noDataMessage.style.display = "block";
                return;
            }
            noDataMessage.style.display = "none";
            data.forEach(item => {
                const row = document.createElement("tr");
                const timeCell = document.createElement("td");
                timeCell.textContent = item.waktu;
                row.appendChild(timeCell);
                const dbCell = document.createElement("td");
                const val = typeof item.desibel === 'object' ? item.desibel.parsedValue :
item.desibel;
                dbCell.textContent = val;
                row.appendChild(dbCell);
                tableBody.appendChild(row);
            });
        })
        .catch(err => console.error('Gagal load history file:', err));
}

function exportToCSV() {
    let csv = "Waktu,Desibel (dB)\n";
    document.querySelectorAll("#historyTableBody tr").forEach(row => {
        const cols = row.querySelectorAll("td");
        if (cols.length === 2) {
            csv += `${cols[0].textContent},${cols[1].textContent}\n`;
        }
    });
    const blob = new Blob([csv], { type: "text/csv" });
    const url = URL.createObjectURL(blob);
    const link = document.createElement("a");
    link.href = url;

```

```

        link.download = "data_suara.csv";
        link.click();
        URL.revokeObjectURL(url);
    }
</script>
</body>
</html>

```

3.2.2 CSS

```

/* Reset dasar */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
/* Layout utama */
body, html {
    height: 100%;
    font-family: Arial, sans-serif;
}
.container {
    display: flex;
    height: 100vh;
}
/* Sidebar */
.sidebar {
    width: 200px;
    background-color: #2c3e50;
    display: flex;
    flex-direction: column;
    padding: 20px;
}
.sidebar button {
    background-color: #34495e;
    color: white;
    border: none;
    padding: 15px;
    margin-bottom: 10px;
    cursor: pointer;
    font-size: 16px;
    border-radius: 5px;
    transition: background 0.3s;
}
.sidebar button:hover {
    background-color: #1abc9c;
}
/* Konten */
.content {
    flex-grow: 1;
    padding: 30px;
}

```

```

    overflow-y: auto;
}
/* Style setiap menu */
.menu {
    display: none;
}
.menu:not(.hidden) {
    display: block;
}
/* Status */
.status-container {
    background-color: #ecf0f1;
    padding: 20px;
    border-radius: 10px;
    text-align: center;
}
.status-aktif {
    color: green;
    font-weight: bold;
}
.status-tidak-aktif {
    color: red;
    font-weight: bold;
}
.button-on, .button-off {
    padding: 10px 20px;
    margin: 10px 5px;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
}
.button-on {
    background-color: #2ecc71;
    color: white;
}
.button-off {
    background-color: #e74c3c;
    color: white;
}
/* Table History */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
table th, table td {
    border: 1px solid #bdc3c7;
    padding: 10px;
    text-align: center;
}

```

```

}
table th {
  background-color: #3498db;
  color: white;
}
.export-btn {
  margin-top: 20px;
  padding: 10px 15px;
  background-color: #2980b9;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.export-btn:hover {
  background-color: #1abc9c;
}
/* Grafik */
#soundChart {
  margin-top: 20px;
}

```

3.2.3 JSS

```

// Fungsi untuk menampilkan menu yang aktif
// =====
function showMenu(menuId) {
  const menus = document.querySelectorAll('.menu');
  menus.forEach(menu => {
    if (menu.id === menuId) {
      menu.classList.remove('hidden'); // Tampilkan menu yang dipilih
    } else {
      menu.classList.add('hidden'); // Sembunyikan menu lain
    }
  });
}
// Fungsi untuk memuat file history yang dipilih
// =====
function loadSelectedHistoryFile() {
  const fileName = document.getElementById("historyFileSelect").value;
  if (fileName) {
    fetch(`/grafik?tanggal=${fileName}`)
      .then(response => {
        if (!response.ok) {
          throw new Error('Gagal memuat data');
        }
        return response.json();
      })
      .then(data => {
        const tableBody = document.getElementById("historyTableBody");
        tableBody.innerHTML = ""; // Hapus data lama

```

```

    if (data.length === 0) {
      const row = document.createElement("tr");
      const cell = document.createElement("td");
      cell.colSpan = 2;
      cell.textContent = "Tidak ada data untuk tanggal ini.";
      row.appendChild(cell);
      tableBody.appendChild(row);
      return;
    }
    data.forEach(item => {
      const row = document.createElement("tr");
      const timeCell = document.createElement("td");
      timeCell.textContent = item.waktu;
      row.appendChild(timeCell);

      const dbCell = document.createElement("td");
      let desibelValue = item.desibel;
      if (typeof desibelValue === 'object' && desibelValue !== null && 'parsedValue' in
desibelValue) {
        desibelValue = desibelValue.parsedValue;
      }
      dbCell.textContent = desibelValue;
      row.appendChild(dbCell);
      tableBody.appendChild(row);
    });
  });
  .catch(err => {
    console.error("Gagal memuat data:", err);
    alert("Gagal memuat data. Silakan coba lagi.");
  });
}
}

// =====
// Fungsi untuk memuat daftar file history dari SPIFFS
// =====
function loadHistoryFiles() {
  fetch("/list")
    .then(response => {
      if (!response.ok) {
        throw new Error('Gagal memuat daftar file');
      }
      return response.json();
    })
    .then(files => {
      const selectElement = document.getElementById("historyFileSelect");
      files.forEach(file => {
        const option = document.createElement("option");
        option.value = file;
        option.textContent = file;

```

```

        selectElement.appendChild(option);
    });
})
.catch(err => {
    console.error("Gagal memuat daftar file:", err);
    alert("Gagal memuat daftar file. Silakan coba lagi.");
});
}
// =====
// Fungsi untuk memperbarui status dan desibel (halaman Beranda)
// =====
function updateStatusAndDesibel() {
    fetch('/status')
    .then(res => res.json())
    .then(data => {
        document.getElementById("statusText").textContent = data.status;
        document.getElementById("desibelNow").textContent = `${data.desibel} dB`;
    })
    .catch(err => {
        console.error("Gagal ambil status:", err);
    });
}
// =====
// Fungsi ON/OFF dari tombol
// =====
function setDeviceState(state) {
    fetch(`/${state}`)
    .then(res => {
        if (res.ok) {
            updateStatusAndDesibel(); // Refresh status setelah ON/OFF
        }
    })
    .catch(err => {
        console.error("Gagal set alat:", err);
    });
}
// =====
// Grafik Real-time (Chart.js)
// =====
let chart;
function initChart() {
    const ctx = document.getElementById("soundChart").getContext("2d");
    chart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: [],
            datasets: [{
                label: 'Desibel (dB)',
                data: [],
                borderColor: 'rgb(75, 192, 192)',

```



```

        fill: false,
        tension: 0.3
    }]
},
options: {
    scales: {
        x: {
            title: { display: true, text: 'Waktu' }
        },
        y: {
            beginAtZero: true,
            title: { display: true, text: 'Desibel (dB)' }
        }
    }
}
});
}

function updateChart() {
    fetch('/data')
    .then(res => res.json())
    .then(data => {
        const time = data.waktu;
        const db = data.desibel;
        if (chart.data.labels.length >= 10) {
            chart.data.labels.shift();
            chart.data.datasets[0].data.shift();
        }

        chart.data.labels.push(time);
        chart.data.datasets[0].data.push(db);
        chart.update();
    })
    .catch(err => console.error("Gagal update chart:", err));
}
// Saat halaman dimuat
document.addEventListener("DOMContentLoaded", function () {
    loadHistoryFiles(); // Untuk menu History
    updateStatusAndDesibel(); // Untuk halaman Beranda
    setInterval(updateStatusAndDesibel, 1000); // Update status & desibel tiap 1 detik
    initChart(); // Inisialisasi grafik
    setInterval(updateChart, 1000); // Update grafik tiap 1 detik
});
// Tambahkan fungsi refresh daftar file
function refreshHistoryFiles() {
    const select = document.getElementById("historyFileSelect");
    select.innerHTML = '<option value="">Pilih tanggal</option>'; // Reset dulu
    loadHistoryFiles();
}
// Update loadSelectedHistoryFile supaya ada loading

```

```

function loadSelectedHistoryFile() {
  const fileName = document.getElementById("historyFileSelect").value;
  const tableBody = document.getElementById("historyTableBody");
  const noDataMessage = document.getElementById("noDataMessage");

  if (!fileName) return;
  fetch(`/grafik?tanggal=${fileName}`)
    .then(res => res.json())
    .then(data => {
      tableBody.innerHTML = "";
      if (data.length === 0) {
        noDataMessage.style.display = "block";
        return;
      }
      noDataMessage.style.display = "none";
      const hourlyData = {};
      const entriesLebih50 = [];
      data.forEach(item => {
        const waktu = item.waktu;
        const jam = waktu.split(":")[0];
        const val = typeof item.desibel === 'object' ? item.desibel.parsedValue :
item.desibel;
        const desibel = Number(val);
        // Rata-rata per jam
        if (!hourlyData[jam]) {
          hourlyData[jam] = [];
        }
        hourlyData[jam].push(desibel);
        // Suara > 50 dB
        if (desibel > 50) {
          entriesLebih50.push({ waktu, desibel });
        }
      });
      // Gabungkan rata-rata per jam + kejadian suara > 50 dB
      const jamUrut = Object.keys(hourlyData).sort();
      jamUrut.forEach(jam => {
        // Baris rata-rata per jam
        const values = hourlyData[jam];
        const avg = (values.reduce((a, b) => a + b, 0) / values.length).toFixed(2);
        const row = document.createElement("tr");
        row.innerHTML = `<td>${jam}:00</td><td>${avg}</td>`;
        tableBody.appendChild(row);
        // Tambahkan baris kejadian > 50 dB di jam yang sama
        entriesLebih50
          .filter(e => e.waktu.startsWith(jam))
          .forEach(e => {
            const subRow = document.createElement("tr");
            subRow.innerHTML = `<td>${e.waktu}</td><td>${e.desibel.toFixed(2)}</td>`;
            tableBody.appendChild(subRow);
          });
      });
    });
}

```

```

    });
  })
  .catch(err => {
    console.error("Gagal memuat data:", err);
    alert("Gagal memuat data. Silakan coba lagi.");
  });
}
// Tambahkan notifikasi kecil setelah download CSV
function exportToCSV() {
  let csv = "Waktu,Desibel (dB)\n";
  document.querySelectorAll("#historyTableBody tr").forEach(row => {
    const cols = row.querySelectorAll("td");
    if (cols.length === 2) {
      csv += `${cols[0].textContent},${cols[1].textContent}\n`;
    }
  });
  const blob = new Blob([csv], { type: "text/csv" });
  const url = URL.createObjectURL(blob);
  const link = document.createElement("a");
  link.href = url;
  link.download = "data_suara.csv";
  link.click();
  URL.revokeObjectURL(url);
}

```

Riwayat Hidup



Dicky Christian Saul, dengan NIM F44120063, lahir di Palu pada tanggal 27 Desember 2001. Penulis adalah anak ketiga dari empat bersaudara dengan nama kedua orang tua yaitu Jhonny Martin Saul dan Sringah Sutardji. Penulis bertempat tinggal di Donggala, Alamat email penulis adalah sauldicky6@gmail.com, dan nomor HP yang dapat dihubungi adalah 085656965990. Penulis menempuh pendidikan formal di SDN 10 Pantoloan tahun 2008 – 2012 dan SDN 2 Kotanagaya 2012 hingga lulus pada tahun 2014, melanjutkan pendidikan di SMP Negeri 1 Donggala dari tahun 2014 hingga lulus pada tahun 2017, dan menyelesaikan pendidikan menengah atas di SMA Negeri 1 Donggala pada tahun 2020. Saat ini, penulis sedang menempuh pendidikan tinggi di Universitas Tadulako dengan mengambil Program Studi Teknik Elektro.