

Implementasi *peer to peer networking* pada *headset* untuk *streaming audio berbasis internet of things*

Rizky Ardiansyah¹, Anita Marselia², Rieke Adriati Wijayanti³, Putri Elfa Masudia⁴,
Abdul Rasyid⁵, Adzikirani⁶, Arinalhaq Fatachul Aziz⁷
e-mail: ¹rizkyardiansyah@polinema.ac.id, ²marsel3102@gmail.com, ³riekeaw@polinema.ac.id,
⁴putri.elfa@polinema.ac.id, ⁵abdul.rasyid@polinema.ac.id, ⁶adzikirani@polinema.ac.id,
⁷arinalmath@polinema.ac.id

^{1,2,3,4,5,6,7}Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

Informasi Artikel

Riwayat Artikel

Diterima 19 Agustus 2025

Direvisi 27 Oktober 2025

Diterbitkan 30 April 2026

Kata kunci:

Audio Streaming

ESP32

Headset

Peer-to-peer

Keywords:

Audio Streaming

ESP32

Headset

Peer-to-peer

ABSTRAK

Penelitian ini mengevaluasi *headset* berbasis IoT untuk komunikasi di lingkungan kampus menggunakan protokol UDP dan ESP32 dalam jaringan *peer-to-peer* untuk *streaming* suara real-time. Hasil pengujian menunjukkan bahwa ketiga *headset* menghasilkan suara dengan intensitas rata-rata 65.62 dBA, nyaman untuk penggunaan sehari-hari, dan lebih efisien dalam memperkuat suara ponsel (65.82 dBA) dibandingkan suara manusia (65.43 dBA). Delay meningkat seiring bertambahnya jarak: 1.15 detik pada 1.5 meter, 3.18 detik pada 3 meter, dan 4.02 detik pada 4.5 meter. Survei pengguna menunjukkan pandangan positif terhadap kualitas suara dan minimal noise, meskipun beberapa melaporkan adanya delay. *Headset* ini terbukti andal dan serbaguna, memberikan pengalaman mendengarkan yang konsisten dan efisien.

ABSTRACT

This study evaluates IoT-based headsets for campus communication using UDP and ESP32 protocols in a peer-to-peer network for real-time voice streaming. Test results show that all three headsets produce sound with an average intensity of 65.62 dBA, comfortable for everyday use, and more efficient in amplifying mobile phone sounds (65.82 dBA) than human voices (65.43 dBA). Delay increases with distance: 1.15 seconds at 1.5 meters, 3.18 seconds at 3 meters, and 4.02 seconds at 4.5 meters. User surveys show positive views on sound quality and minimal noise, although some report delays. The headsets prove to be reliable and versatile, providing a consistent and efficient listening experience.

Penulis Korespondensi:

Rizky Ardiansyah,

Jurusan Teknik Elektro,

Politeknik Negeri Malang,

Jl. Soekarno Hatta No 9, Malang, Jawa Timur, Indonesia.

Email: rizkyardiansyah@polinema.ac.id

1. PENDAHULUAN

Pada era modern, bahasa memiliki peran penting dalam memahami dan menyampaikan pesan serta menjaga hubungan sosial antar manusia. Seiring dengan kemajuan teknologi di era digital, kebutuhan terhadap komunikasi dan hiburan berbasis teknologi semakin meningkat. Salah satu teknologi yang berkembang pesat adalah Internet of Things (IoT), yang memungkinkan perangkat saling berkomunikasi dan bertukar data secara efisien tanpa intervensi manusia. Perkembangan ini turut mendorong munculnya berbagai alat komunikasi radio dengan beragam merek, keunggulan, dan kekurangannya masing-masing [1]. Namun, di balik

perkembangan tersebut, masih terdapat permasalahan dalam hal komunikasi nirkabel yang efisien, terutama dalam konteks *headset* audio streaming. Banyak perangkat komunikasi masih bergantung pada server pusat yang dapat menyebabkan latensi tinggi, biaya operasional besar, serta keterbatasan jangkauan dan fleksibilitas jaringan. Selain itu, dalam sistem pengiriman suara, faktor seperti media penghubung dan penempatan sumber serta penerima suara sangat memengaruhi kualitas audio yang dihasilkan [2].

Dalam mendapatkan kualitas suara baik perlu di perhatikan beberapa aspek yang mempengaruhi kualitas suara seperti media penghubung dan penempatan sumber suara dan penerima suara[3]. Pada penelitian "Pengembangan Metode Sistem Terdistribusi (Peer to Peer and Client Server) Untuk Informasi Hasil Pertanian Menggunakan IoT" oleh Fredy Susanto, Diah Aryani, dan Ari Asmawati (2020) menggunakan sistem terdistribusi untuk mendistribusikan data hasil panen kepada petani secara real-time melalui server web Arduino Uno dan smartphone[4]. Adapun penelitian lain yang berjudul "Implementasi Perbandingan Sistem Pengiriman Sinyal Suara secara Wireless pada Arduino" oleh Muhammad Rasyid Perdana, Wijaya Kurniawan, dan Rakhmadhany Primananda (2019) mengirimkan sinyal suara melalui WiFi menggunakan ESP8266, diproses oleh Arduino Uno, dan dikirim ke speaker penerima.

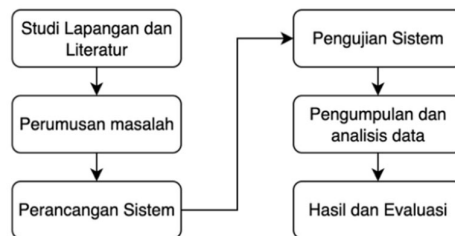
Untuk mengatasi hal tersebut, solusi yang diusulkan dalam penelitian ini adalah implementasi teknologi Peer to Peer (P2P) pada *headset* streaming berbasis IoT. Jaringan P2P memungkinkan dua perangkat atau lebih untuk berkomunikasi langsung tanpa memerlukan server pusat [5]. Pendekatan ini dapat mengurangi latensi, meningkatkan kecepatan transmisi data, dan menurunkan biaya operasional. Sistem yang dirancang menggunakan ESP32 sebagai transmitter dan receiver suara, dengan protokol komunikasi UDP untuk mendukung pengiriman data audio secara real-time. UDP merupakan protokol sederhana yang bersifat tanpa koneksi dan tidak menjamin pengiriman data sepenuhnya [6], namun cocok untuk kebutuhan transmisi cepat seperti streaming suara.

Dengan implementasi ini, diharapkan *headset* mampu mentransmisikan suara secara real-time antarperangkat secara langsung, tanpa bergantung pada infrastruktur jaringan yang kompleks. Selain memberikan efisiensi dalam komunikasi audio, sistem ini juga berpotensi menjadi solusi inovatif di lingkungan kerja kolaboratif, karena memungkinkan komunikasi dua arah dengan latensi rendah dan kualitas suara yang tetap stabil [7]. Penelitian ini diharapkan dapat menjadi dasar pengembangan teknologi komunikasi nirkabel berbasis IoT yang lebih efektif, hemat sumber daya, dan mudah diimplementasikan di masa mendatang.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

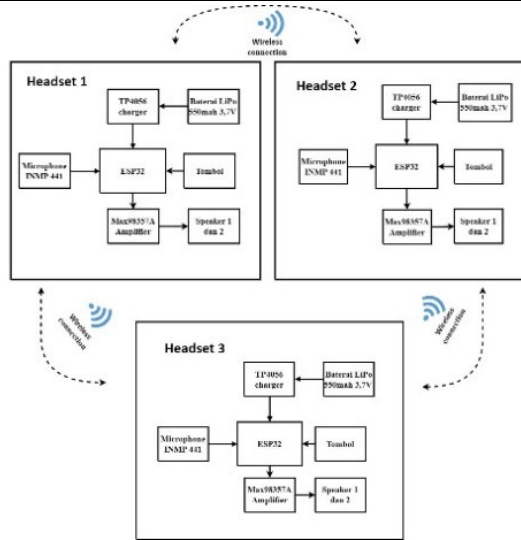
Tahapan penelitian yang akan dilakukan dalam Implementasi *Peer to Peer* Networking pada *Headset* untuk *Streaming Audio* Berbasis *Internet of Things* ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

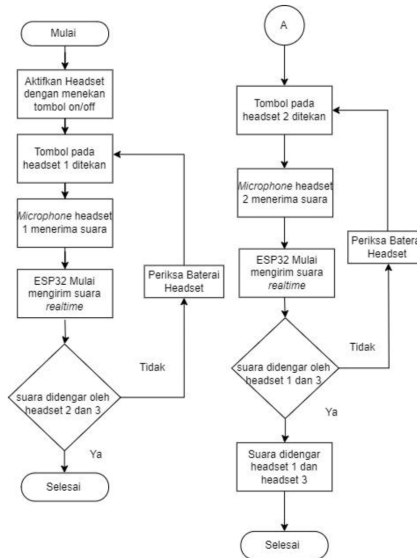
Perancangan alat berisikan Perancangan fisik dari *headset*, Hal ini termasuk bentuk, tata letak komponen-komponen di dalamnya. Arsitektur sistem dijelaskan melalui diagram blok yang menggambarkan interaksi antar komponen utama, serta protokol komunikasi untuk jaringan peer-to-peer. Terdapat tiga *headset* yang terhubung ke ESP32 menggunakan kabel untuk menangkap suara melalui mikrofon INMP441.

Gambar 2 menunjukkan diagram blok dari sistem komunikasi untuk tiga *headset* yang terhubung secara wireless. Setiap *headset* dilengkapi dengan beberapa komponen utama: Microphone INMP441 untuk menangkap suara dan mengubahnya menjadi sinyal digital, serta ESP32 sebagai mikroprosesor yang mengelola fungsi *headset*, memproses sinyal suara, dan mengirimkannya secara wireless. ESP32 juga mengontrol tombol dan berhubungan dengan amplifier serta speaker. Max98357A digunakan untuk menguatkan sinyal audio dari ESP32 sebelum dikirim ke speaker, sementara Speaker 1 dan 2 mengeluarkan suara yang telah diproses dan diperkuat. TP4056 Charger dan Baterai mengatur pengisian daya dan menyediakan tenaga untuk *headset*. Tombol pada *headset* mengaktifkan mikrofon dan mengirimkan sinyal suara saat ditekan lama. Sistem ini memungkinkan komunikasi dua arah secara real-time antara *headset* melalui koneksi wireless.



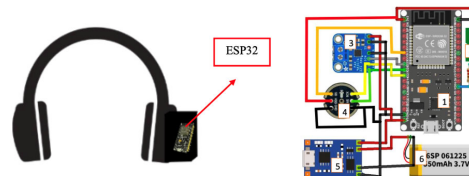
Gambar 2. Blok Diagram Sistem

Suara dapat dikirimkan antar *headset* bergantung pada kondisi tombol yang ditekan pada setiap *headset*. Ketika lebih dari satu *headset* mencoba mengirim suara (pengirim) pada saat yang sama, maka pengiriman suara gagal. Karena pada protokol menggunakan operasi broadcast dan multicast, di mana paket data dapat dikirim ke banyak penerima secara bersamaan dengan hanya mengirim satu data saja. Berikut skenario ada pada Gambar 3.



Gambar 3. Flowchart Sistem

Pada Gambar 4 menunjukkan desain *headset* yang terdapat ESP32 disetiap sisi *headset*nya. ESP32 juga terdapat modul tambahan yaitu mikrofon INMP441 disini berfungsi untuk menghasilkan output *audio* yang jernih dengan *distorsi* rendah. Kemudian akan dikirimkan ke *speaker* penerima. Pada Gambar 4 menyajikan diagram sirkuit terperinci dari *headset* beserta pin yang digunakan.



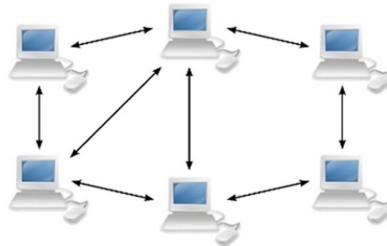
Gambar 4. Desain *headset* dan wiring

2.2 Peer To Peer Networking

Jaringan peer-to-peer (P2P) adalah jaringan komputer di mana setiap komputer dapat saling berbagi sumber daya dan layanan tanpa melalui server pusat. Untuk membuat jaringan P2P, cukup menggunakan kabel UTP yang dipasangkan pada kartu jaringan masing-masing komputer[8]. Bahkan untuk membuat jaringan peer-to-peer dengan dua computer tidak perlu menggunakan hub atau switch, namun cukup menggunakan 1 kabel UTP yang dipasangkan pada kartu jaringan masing masing computer[9].

Dalam jaringan P2P, setiap node dapat berfungsi sebagai klien atau server:

- Sebagai Klien: Meminta data dari node lain, Menginisiasi koneksi untuk bertukar data.
- Sebagai Server: Menyediakan dan membagikan data kepada node lain, Menerima koneksi dan merespons permintaan data atau layanan[10].



Gambar 5. Peer to Peer Network Topology

2.3 Audio Streaming

Streaming merupakan sebuah teknologi yang memungkinkan pengguna untuk melakukan pemutaran audio atau video yang dilakukan secara langsung (live) maupun tidak langsung (prerecord) di web server tanpa dilakukan proses download sampai selesai terlebih dahulu[11]. Streaming memungkinkan pengiriman data suara atau video yang sudah dikompresi melalui internet secara real-time, tanpa perlu mengunduh atau menyimpan file secara lokal. Streaming audio memungkinkan pengiriman konten audio secara terus-menerus dan real-time, sehingga pengguna dapat mendengarkan audio saat data masih dalam proses pengunduhan[12].

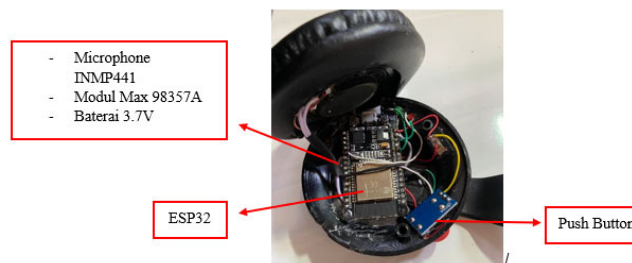
2.4 User Data Protocol

UDP (User Datagram Protocol) adalah protokol komunikasi pada lapisan transport dalam model OSI dan TCP/IP. Sebagai protokol tanpa koneksi, UDP memungkinkan pengiriman datagram data dengan efisiensi tinggi, mendukung komunikasi yang tidak andal dan tanpa koneksi antara host dalam jaringan TCP/IP. Cara kerja UDP melibatkan dua proses utama yaitu pengiriman dan penerimaan data. Pada pengiriman data, aplikasi pengirim membentuk datagram dengan menambahkan data ke header UDP, kemudian datagram tersebut dikirim ke lapisan IP untuk diteruskan ke penerima. Pada penerimaan data, datagram yang diterima oleh lapisan IP diserahkan ke lapisan UDP, lalu aplikasi penerima menerima dan memproses datagram tersebut. Kedua protokol ini mempunyai kelebihan dan kekurangan masing-masing tergantung tujuan yang ingin dicapai, dimana kelebihan TCP yaitu reliable pada datanya, sedangkan UDP tidak memerlukan memori yang besar dalam pengirimannya[13][14][15].

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pembuatan Alat

Gambar 3 dan 4 menunjukkan keseluruhan sistem yang telah dirancang berupa 3 buah *headset*. Alat yang di rancang meliputi *Headset*, Microphone INMP441, Modul Max 98357A, Baterai 3.7 V, ESP32, dan Push Button.



Gambar 6. Komponen didalam *Headset*



Gambar 7. Headset

3.2. Hasil Uji Intensitas Suara

Pengujian intensitas suara bertujuan untuk memastikan bahwa *headset* atau perangkat audio tidak mengeluarkan suara yang terlalu keras dan dapat membahayakan pendengaran pengguna.

Gambar 8. Pengujian menggunakan *Sound Level Meter*

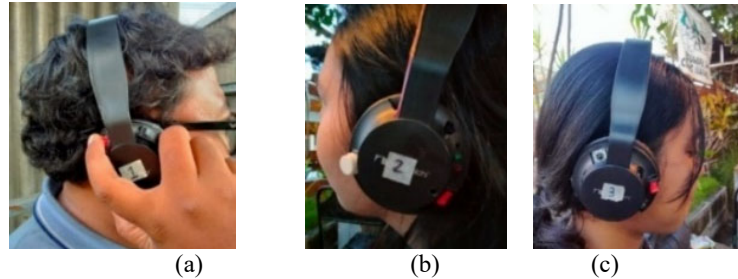
Tabel 1. Hasil Uji Intensitas Suara

No.	Sumber Suara	Jarak (m)	Tidak ada Suara (dBA)	Terdengar Suara (dBA)			Rata – Rata (dBA)
				H1	H2	H3	
1.	Manusia	1.5 m	33.5	66.1	66.9	66.2	66.4
2.	Manusia	3 m	33.5	65.7	65.3	65.8	65.6
3.	Manusia	5 m	33.5	64.1	64.5	64.3	64.3
4.	Speaker Handphone	1.5 m	33.5	66.2	66.7	66.9	66.6
5.	Speaker Handphone	3 m	33.5	66.0	66.1	65.3	65.8
6.	Speaker Handphone	5 m	33.5	64.9	65.1	65.2	65.06

Tingkat intensitas suara bervariasi sedikit antara posisi H1, H2, dan H3, menunjukkan distribusi suara yang seragam di sekitar telinga, Intensitas dengan rata-rata 65.50 dBA, 65.76 dBA, dan 66.61 dBA. Suara dari speaker ponsel selalu lebih intens dibandingkan suara manusia, dengan rata-rata 65.82 dBA untuk speaker ponsel dan 65.43 dBA untuk suara manusia. Saat tidak bersuara, intensitas manusia dan speaker ponsel sama-sama 33.5 dBA, dan saat berbicara, manusia berkisar antara 64.1 dBA hingga 66.9 dBA, sedangkan speaker ponsel antara 64.9 dBA hingga 66.9 dBA. *Headset* terbukti efektif memperkuat audio eksternal dengan rata-rata intensitas 65.82 dBA dari speaker ponsel dan 65.43 dBA dari suara manusia. Rata-rata intensitas suara keseluruhan dalam berbagai kondisi adalah 65.62 dBA. Nilai ini menunjukkan bahwa *headset* menghasilkan tingkat intensitas suara sedang, berada dalam kisaran percakapan umum. Dengan rata-rata tingkat intensitas suara 65.62 dBA, *headset* menunjukkan kemampuan yang memadai untuk menghasilkan suara yang jelas dan dapat didengar dalam berbagai situasi. *Headset* ini cocok digunakan untuk percakapan dan streaming audio, karena tingkat suaranya berada dalam rentang yang nyaman bagi telinga manusia[11].

3.3. Hasil Pengujian Peer to Peer

Pengujian peer to peer dilakukan untuk mengetahui apakah suara antara *headset* 1 dengan *headset* 2 dan *headset* 3 dapat terhubung secara bergantian dan masing-masing *headset* memiliki fungsi yang sama sebagai pengirim dan penerima pengujian peer to peer ditunjukkan pada Gambar 6.



(a) (b) (c)
Gambar 6. Pengujian *Peer to Peer*

Tabel 2. Hasil Uji *Peer To Peer*

No.	Headset 1	Headset 2	Headset 3	Hasil
1.	Berbicara	Mendengar	Mendengar	Suara yang dikirimkan dari <i>headset 1</i> terdengar oleh <i>headset 2</i> dan <i>headset 3</i>
2.	Mendengar	Berbicara	Mendengar	Suara yang dikirimkan dari <i>headset 2</i> terdengar oleh <i>headset 1</i> dan <i>headset 3</i>
3.	Mendengar	Mendengar	Berbicara	Suara yang dikirimkan dari <i>headset 3</i> terdengar oleh <i>headset 1</i> dan <i>headset 2</i>

Hasil pengujian menunjukkan bahwa pada setiap percobaan, hanya satu *headset* yang diaktifkan dengan menekan tombol dalam waktu lama, sedangkan dua *headset* lainnya hanya menerima suara. Suara yang dikirim dari *headset* aktif dapat didengar oleh kedua *headset* pasif. Pengujian ini menunjukkan bahwa koneksi peer-to-peer antara ketiga *headset* berhasil, karena setiap kali satu *headset* mengirim suara, suara tersebut diterima oleh dua *headset* lainnya. Menekan tombol dalam waktu lama pada masing-masing *headset* berfungsi untuk mengaktifkan transmisi suara, membuktikan bahwa sistem komunikasi antar-*headset* berfungsi dengan baik.

3.4. Hasil Pengujian Delay dan Jarak

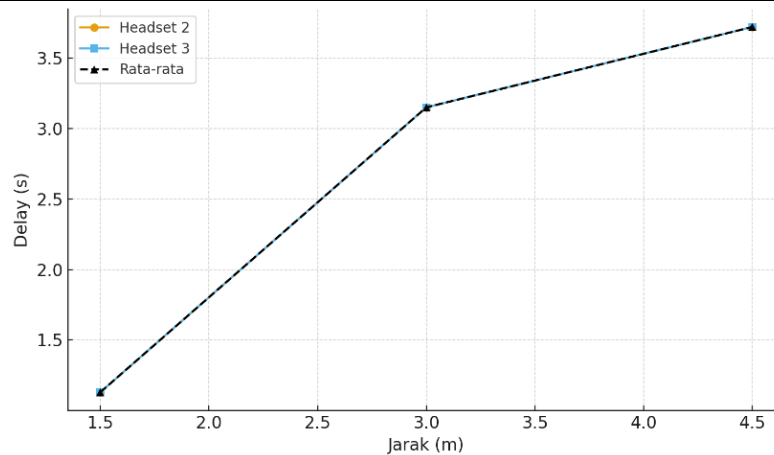
Pengujian delay pada setiap jarak digunakan untuk mengukur dan menganalisis waktu tunda (delay) yang terjadi saat sinyal atau data ditransmisikan dari satu titik ke titik lainnya dengan menggunakan stopwatch untuk menentukan waktunya.

Tabel 3. Hasil Pengujian Delay dan jarak

No.	Jarak(m)	Nilai Percobaan		Rata – Rata (s)
		Headset 2 (s)	Headset 3 (s)	
1	1.5 m	01.13	01.13	01.13
2	3.5 m	03.15	03.15	03.15
3	4.5 m	03.72	03.72	03.72

Pada Tabel 3 menunjukkan hasil percobaan untuk mengukur waktu respon *headset* pada tiga jarak: 1,5meter, 3 meter, dan 4,5 meter. Pengukuran dilakukan pada tiga *headset*, dengan hasil rata-rata sebagai berikut:

- Jarak 1.5 meter, semua *headset* mampu memberikan waktu respon yang cukup cepat dan konsisten dengan nilai rata-rata mencapai 01.13 detik, yang mengindikasikan bahwa sistem dapat berfungsi dengan sangat baik dan stabil pada jarak dekat tanpa adanya penurunan performa yang signifikan.
- Jarak 3meter terjadi peningkatan delay sedikit dengan rentang antara 0.3 hingga 0.32 detik dan rata-rata 0.315 detik dari 5 percobaan. Variasi kecil mulai terlihat antara *headset* yang digunakan. Meskipun begitu, waktu respon masih rendah, menunjukkan bahwa sistem tetap berfungsi baik pada jarak menengah, meski ada peningkatan delay.
- Pada jarak 4.5meter, delay meningkat signifikan dengan rentang 0.364 hingga 0.379 detik dan rata-rata 0.372 detik. Variasi antar *headset* juga meningkat, menunjukkan bahwa pada jarak lebih jauh, sistem mengalami peningkatan delay dan performa antar perangkat bisa bervariasi.



Gambar 7. Grafik Perbandingan Delay Headset pada berbagai jarak

Hasil ini menunjukkan bahwa *headset* berfungsi baik pada jarak dekat. Artinya, peningkatan jarak berbanding lurus dengan peningkatan waktu respon sistem

4. KESIMPULAN

Ketiga *headset* menghasilkan suara dengan intensitas rata-rata 65.62 dBA, nyaman untuk penggunaan sehari-hari. *Headset* lebih efisien dalam memperkuat suara dari pengeras suara ponsel (65.82 dBA) dibandingkan suara manusia (65.43 dBA), dan efektif menangkap suara manusia dengan rentang 64.1 hingga 66.9 dBA. Secara keseluruhan, *headset* ini andal dan serbaguna, memberikan pengalaman mendengarkan yang konsisten dan efisien.

Waktu respon *headset* bervariasi dengan jarak: pada 1,5meter, delay cukup cepat (0.113 detik) dan stabil; pada 3 meter, meningkat signifikan menjadi 0.315 detik dengan variasi kecil antar *headset*; dan pada 4.5meter, delay sedikit meningkat menjadi 0.372 detik dengan variasi yang lebih besar. Ini menunjukkan bahwa performa *headset* dipengaruhi oleh jarak, dengan delay dan variasi performa meningkat seiring bertambahnya jarak.

5. DAFTAR PUSTAKA

- [1] M. T. Prakarsa, D. Wahyuni, N. Rachman, and I. Mujahidin, "Optimasi sistem komunikasi dari HT dengan HP dalam pelaksanaan tugas operasi TNI AD menggunakan metode DTMF," *JASIEK*, vol. 1, no. 1, p. 57–65, Jun. 2019, doi: 10.26905/jasiek.v1i1.3150.
- [2] A. M. Shiddiqi, A. Z. Abidin, and W. Wibisono, "Rancang bangun jaringan *peer to peer* dengan konsep semantic overlay networks," *Seminar Nasional Informatika 2010 (semnasIF 2010)*, pp. 9–14, 2010.
- [3] P. V. B. Romony, L. Sitanayah, and J. B. Sanger, "Perbandingan *quality of service* protokol komunikasi data pada sistem deteksi asap rokok berbasis *internet of things*," *realtech*, vol. 16, no. 1, pp. 19–24, Apr. 2020, doi: 10.52159/realtech.v16i1.129.
- [4] M. R. Perdana, W. Kurniawan, and R. Primananda, "Implementasi Perbandingan Sistem Pengiriman Sinyal Suara secara Wireless pada Arduino".
- [5] F. Susanto, D. Aryani, and A. Asmawati, "Pengembangan Metode Sistem Terdistribusi (*Peer to Peer* and *Client Server*) Untuk Informasi Hasil Pertanian Menggunakan IOT".
- [6] S. Haritha and R. A. Reddy, "Design and Implementation of Efficient Audio Streaming System Using Raspberry Pi," *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, vol. 12, no. 1, pp. 33–36, Dec. 2014.
- [7] T. S. Kalengkongan and D. J. Mamahit, "Rancang Bangun Alat Deteksi Kebisingan Berbasis Arduino Uno," vol. 7, no. 2, pp. 186–188, 2018.
- [8] R. D. Saputra, "implementasi jaringan *peer to peer* dalam proses transfer data dua *personal computer* menggunakan kabel UTP bertipe *cross*," pp. 1–22.
- [9] J. Suwardinata, D. Prawira, and F. Febriyanto, "Sistem Radio Streaming Berbasis Android Studi Kasus: Panji TV Pontianak," *Coding j. komp'ut. dan aplikasi*, vol. 10, no. 01, p. 60, May 2022, doi: 10.26418/coding.v10i01.52480.
- [10] J. D. Wicaksono, W. Kurniawan, and M. H. H. Ichsan, "Implementasi Metode Pervasive Menggunakan Protokol UDP Pada Raspberry Pi dan myRIO".
- [11] T. Peneliti et al., "analisis pengukuran taraf intensitas bunyi kendaraan bermotor di SMP Mardisiswa 2 Semarang".
- [12] A. Rahman, M. Firmansyah, and R. S. Pradana, "Design and Implementation of Wireless Audio Transmission Using ESP32 with UDP Protocol," *Jurnal Teknologi dan Sistem Komputer*, vol. 10, no. 2, pp. 115–122, 2022.

-
- [13] S. K. Sharma and P. Singh, “*IoT Based Peer-to-Peer Communication Framework for Real-Time Applications*,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 4, pp. 458–465, 2023.
 - [14] L. T. Handayani, D. Purnomo, and A. H. Nugroho, “*Performance Evaluation of UDP and TCP Protocols for Audio Streaming over IoT Network*,” Procedia Computer Science, vol. 207, pp. 234–241, 2022.
 - [15] F. Santoso and M. R. Permana, “*Development of Smart Headset Communication System Based on ESP32 and Wi-Fi Direct*,” IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), pp. 1–6, 2021.