



IMPLEMENTASI PROTOKOL TRANSMISI DATA AKUSTIK ZERO-INSTALL BERBASIS WEB AUDIO API DAN WEBASSEMBLY UNTUK KOMUNIKASI PADA LINGKUNGAN AIR-GAPPED

Eko Prasetyo Dilapanga ^{1*}, Miracle Patric Lumowa ², Roynaldo Christofel Johaness Mailangkay ³, Sherwin Reinaldo U Sompie ⁴, Wahyuni Fithratul Zahmi ⁵, Brave Angkasa Sugiarso ⁶, Ade Yusupa ⁷

^{1,2,3,4,5,6,7}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi
Kota Manado, Indonesia

E-mail: ekodilapanga026@student.unsrat.ac.id^{1*}

miraclelumowa026@student.unsrat.ac.id^{2*}

roynaldomailangkay026@student.unsrat.ac.id^{3*}

aldo@unsrat.ac.id^{4*}

wahyuni.fithratul.zalmi@unsrat.ac.id^{5*}

brave@unsrat.ac.id^{6*}

ade@unsrat.ac.id^{7*}

ABSTRACT

Air-gapped environments in critical infrastructure require secure data transfer methods without violating physical isolation principles. Conventional methods such as USB storage are vulnerable to malware, while existing acoustic transmission solutions are generally based on native applications that require installation, often conflicting with strict security policies. This study aims to implement an acoustic data transmission protocol based on Web Audio API and WebAssembly (WASM), enabling real-time encrypted text communication via web browsers without requiring software installation (zero-install). The system was developed using the ggwave library for Frequency Shift Keying (FSK) signal modulation and tested for performance based on latency, data integrity, and physical distance parameters. The results demonstrate that the system operates effectively across platforms. The fastest transmission protocol offers 40% lower latency compared to the standard protocol but experiences a significant increase in Character Error Rate (CER) under noisy conditions. The system proves optimal as a near-field communication method with a 100% success rate at distances under 50 cm. This study concludes that the web-based approach is valid as a practical and secure communication bridge solution for air-gapped environments.

Keywords: Acoustic Communication, Web Audio API, WebAssembly, Air-Gapped, Zero-Install

ABSTRAK

Lingkungan sistem terisolasi (*air-gapped*) pada infrastruktur kritis memerlukan metode transfer data yang aman tanpa melanggar prinsip isolasi fisik. Metode konvensional seperti media

Article History

Received: Desember 2025

Reviewed: Desember 2025

Published: Desember 2025

Plagiarism Checker No
235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



penyimpanan USB rentan terhadap *malware*, sedangkan solusi transmisi akustik yang ada umumnya berbasis aplikasi natif yang memerlukan instalasi, sehingga sering kali terkendala kebijakan keamanan yang ketat. Penelitian ini bertujuan mengimplementasikan protokol transmisi data akustik berbasis Web Audio API dan WebAssembly (WASM) yang memungkinkan komunikasi teks terenkripsi secara *real-time* melalui peramban web tanpa memerlukan instalasi perangkat lunak (*zero-install*). Sistem dikembangkan menggunakan pustaka *ggwave* untuk modulasi sinyal *Frequency Shift Keying* (FSK) dan diuji kinerjanya berdasarkan parameter latensi, integritas data, dan jarak fisik. Hasil pengujian menunjukkan bahwa sistem mampu beroperasi secara lintas platform (*cross-platform*) dengan efektif. Protokol transmisi cepat (*fastest protocol*) menawarkan latensi 40% lebih rendah dibandingkan protokol standar, namun mengalami peningkatan tingkat kesalahan karakter (*Character Error Rate*) yang signifikan pada kondisi bising. Sistem terbukti bekerja optimal sebagai metode komunikasi jarak dekat (*near-field*) dengan tingkat keberhasilan 100% pada jarak di bawah 50 cm. Penelitian ini menyimpulkan bahwa pendekatan berbasis web valid sebagai solusi jembatan komunikasi praktis dan aman untuk lingkungan *air-gapped*.

Kata Kunci: Komunikasi Akustik, Web Audio API, WebAssembly, Air-Gapped, Zero-Install

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Di era hiperkonektivitas *Internet of Things* (IoT), pertukaran data nirkabel didominasi oleh protokol frekuensi radio (RF) seperti Wi-Fi dan Bluetooth. Namun, dalam infrastruktur kritis seperti sistem militer, pembangkit listrik, atau penyimpanan aset digital (*cold wallets*), konektivitas standar ini justru menjadi celah keamanan. Oleh karena itu, diterapkan mekanisme *air-gapped*, yaitu isolasi fisik total dari jaringan publik maupun lokal untuk mencegah intrusi siber. Tantangan fundamental pada lingkungan *air-gapped* adalah kebutuhan mentransfer data esensial (seperti kunci enkripsi atau konfigurasi) tanpa melanggar prinsip isolasi tersebut. Penggunaan media fisik seperti USB flash drive terbukti berisiko tinggi terhadap penyebaran *malware*, sebagaimana dibuktikan oleh kasus Stuxnet. Sebagai alternatif, metode komunikasi data akustik (*data-over-sound*) muncul sebagai solusi menjanjikan karena memanfaatkan perangkat keras *ubiquitous* (speaker dan mikrofon) yang sudah tersedia pada hampir semua perangkat komputasi modern tanpa perlu tambahan perangkat keras khusus.



Meskipun potensial, implementasi transmisi akustik saat ini memiliki kelemahan signifikan yaitu mayoritas solusi berbasis pada aplikasi natif (*native apps*). Hal ini menciptakan friksi bagi pengguna dan tantangan kompatibilitas lintas platform. Lebih krusial lagi, pada lingkungan dengan keamanan tinggi, kebijakan TI seringkali melarang instalasi perangkat lunak pihak ketiga, sehingga solusi berbasis aplikasi natif menjadi tidak praktis. Terobosan teknologi web modern melalui HTML5 dan Web Audio API kini memungkinkan pemrosesan sinyal audio secara *real-time* di peramban. Ditambah dengan integrasi WebAssembly (WASM), kode tingkat rendah seperti C++ dapat dijalankan di web dengan performa mendekati *native*. Namun, terdapat kesenjangan penelitian (*research gap*) di mana studi yang ada lebih berfokus pada analisis teoritis kapasitas kanal audio, dan masih sedikit yang mengeksplorasi implementasi praktis protokol akustik yang sepenuhnya berjalan di *browser* untuk sistem *air-gapped*. Tantangan teknis seperti manajemen AudioContext, konversi *sample rate*, dan integrasi WASM untuk enkripsi *real-time* belum dibahas secara komprehensif.

Berdasarkan urgensi tersebut, penelitian ini bertujuan mengimplementasikan protokol transmisi data akustik berbasis Web Audio API menggunakan pustaka *ggwave* yang dikompilasi ke WebAssembly. Pendekatan ini menawarkan mekanisme komunikasi *device-to-device* yang aman, lintas platform, dan bersifat *zero-install*, menjadikannya solusi jembatan komunikasi yang efektif dan patuh pada restriksi keamanan lingkungan *air-gapped*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah:

- Bagaimana merancang mekanisme enkripsi dan dekripsi sinyal audio berbasis pustaka *ggwave* ke dalam lingkungan *browser* menggunakan Web Audio API dan WebAssembly untuk memungkinkan komunikasi tanpa instalasi (*zero-install*)?
- Bagaimana perbandingan kinerja transmisi data antara protokol standar dan protokol cepat (*fastest protocol*) pada platform web ditinjau dari latensi pemrosesan dan akurasi penerimaan pesan?
- Seberapa tangguh sistem komunikasi akustik berbasis web ini dalam mempertahankan integritas data pada lingkungan *air-gapped* ketika menghadapi variasi jarak fisik dan gangguan kebisingan latar belakang (*ambient noise*)?



1.3 Rumusan Masalah

Tujuan dari penelitian ini adalah:

- Merancang dan membangun aplikasi web menggunakan React, TypeScript, dan WebAssembly (WASM) yang mampu melakukan encoding dan decoding sinyal audio secara *real-time* tanpa instalasi perangkat lunak tambahan.
- Mengukur secara kuantitatif parameter latensi pemrosesan (*processing overhead*) dan integritas data (persentase keberhasilan decode) pada konfigurasi protokol yang berbeda.
- Menguji batas operasional sistem dalam skenario fisik *air-gapped* dengan mengukur tingkat keberhasilan transmisi (*success rate*) pada variasi jarak dan kondisi kebisingan.

2. TINJAUAN PUSTAKA

2.1 Kajian Teori

Sub-bab ini menguraikan landasan teoritis yang menjadi fondasi penelitian, dibagi menjadi konsep keamanan jaringan dan teknologi pemrosesan sinyal berbasis web.

2.1.1 Keamanan Sistem Terisolasi (*Air-Gapped Systems*)

Keamanan infrastruktur kritis seringkali bergantung pada isolasi fisik. Menurut National Institute of Standards and Technology (NIST), air gap didefinisikan sebagai antarmuka antara dua sistem di mana mereka tidak terhubung secara fisik dan tidak berbagi mekanisme komunikasi logis, bertujuan mencegah penyusupan atau kebocoran data. Dalam konteks ini, tantangan utamanya adalah memindahkan data melintasi celah udara tanpa melanggar prinsip isolasi tersebut.

2.2.2 Komunikasi Akustik dan Modulasi FSK

Sebagai solusi jembatan komunikasi, digunakan teknologi *Data-over-Sound* (DoS) yang memanfaatkan gelombang suara sebagai pembawa (*carrier*). Teknik modulasi yang diterapkan adalah Frequency Shift Keying (FSK), yaitu merepresentasikan data biner dengan menggeser frekuensi sinyal di antara kumpulan frekuensi yang telah ditentukan. FSK dipilih karena karakteristiknya yang memiliki ketahanan (*robustness*) yang baik terhadap gangguan amplitudo (*noise*) di udara.

2.2.3 Teknologi Web Modern: Web Audio API dan WebAssembly

Untuk mengatasi kendala instalasi, penelitian ini memanfaatkan standar web modern:

- Web Audio API: Spesifikasi W3C yang memungkinkan pemrosesan dan sintesis audio secara langsung di peramban dengan presisi waktu tinggi. Ini memungkinkan akses mikrofon dan speaker tanpa plugin tambahan.
- WebAssembly (WASM): Format instruksi biner yang memungkinkan kode bahasa tingkat rendah (C/C++) berjalan di peramban dengan kecepatan mendekati aplikasi asli (*native speed*). Sintesis: Peran WASM sangat krusial dalam penelitian ini karena algoritma DSP (seperti modulasi FSK pada pustaka *ggwave*) memerlukan komputasi berat yang jika dijalankan dengan JavaScript murni akan mengalami masalah latensi dan *jitter*. Integrasi ini menjembatani kesenjangan performa antara aplikasi web dan desktop.

2.2 Penelitian Terdahulu

Tinjauan literatur menunjukkan adanya evolusi dalam komunikasi data akustik yang dapat dikategorikan menjadi tiga domain:

- Eksfiltrasi Data (Perspektif Ofensif): Penelitian oleh Guri dkk. mendemonstrasikan kelayakan transmisi data melalui celah udara (misalnya Fansmitter), namun mayoritas berfokus pada teknik *malware* satu arah (*one-way*) untuk pencurian data dan memerlukan akses sistem tingkat rendah.
- Protokol Berbasis Aplikasi Natif: Solusi komersial dan akademis seperti Chirp.io menggunakan modulasi FSK untuk pertukaran data jarak dekat. Kelemahannya adalah kewajiban instalasi aplikasi (*native installation*), yang sering kali dilarang dalam kebijakan keamanan TI yang ketat.
- Implementasi DSP di Web: Meskipun Web Audio API mampu memproses audio, implementasi awal yang menggunakan JavaScript murni sering mengalami *bottleneck* kinerja saat menangani enkripsi *real-time*, mengakibatkan tingginya tingkat kesalahan data.



Penelitian ini mengisi kesenjangan (*research gap*) dengan menghadirkan solusi komunikasi dua arah yang aman, namun berjalan sepenuhnya di lingkungan peramban (*browser-based*) tanpa instalasi. Perbandingan fitur utama disajikan dalam Tabel 2.1.

Tabel 2.1. Matriks Perbandingan Kebaruan Penelitian

Parameter	Penelitian Terdahulu (Umum)	Penelitian Ini (GibberType)
Platform	Aplikasi Natif (Android/iOS/Desktop) atau <i>Malware</i> level OS	<i>Web Browser</i> Modern (Chrome, Edge, Firefox, Safari)
Instalasi	Memerlukan instalasi perangkat lunak (<i>Install Required</i>)	Tanpa Instalasi (<i>Zero-Install</i>) via URL
Teknologi Inti	C/C++ <i>Native</i> atau Java/Swift	Web Audio API + WebAssembly (WASM)
Fokus Utama	Eksfiltrasi data (pencurian) atau <i>pairing</i> perangkat IoT	Komunikasi Teks Dua Arah Aman & Transparan
Visualisasi	Umumnya tidak ada atau statis	<i>Real-time Spectrum Analysis</i> (menggunakan AudioMotion-Analyzer)



2.3 Kerangka Berpikir

Kerangka berpikir penelitian ini dibangun berdasarkan alur pemecahan masalah sistem *air-gapped* melalui pendekatan rekayasa perangkat lunak.

- Input (Identifikasi Masalah): Kebutuhan transfer data pada sistem terisolasi terkendala oleh risiko fisik (USB mengandung *malware*) dan risiko radio (Wi-Fi/Bluetooth rentan sadap).
- Proses (Solusi Teknologi): Menggunakan pendekatan berbasis web untuk mengatasi masalah kompatibilitas.
 - Web Audio API berfungsi sebagai antarmuka I/O untuk mikrofon dan speaker.
 - WebAssembly (WASM) menjalankan pustaka *ggwave* untuk proses encoding (Teks → Sinyal FSK) dan decoding (Sinyal FSK → Teks) secara efisien.
- Output (Variabel Penelitian): Kinerja sistem yang dibangun diukur berdasarkan variabel berikut:
 - Variabel Bebas: Protokol Transmisi (Standard vs Fastest), Jarak Fisik (10cm - 200cm), dan Kondisi Lingkungan (Hening vs Bising).
 - Variabel Terikat: Tingkat Keberhasilan Karakter (Character Error Rate/CER) dan Latensi Sistem.

3. METODOLOGI PENELITIAN

3.1 Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan jenis Penelitian Terapan (*Applied Research*) dengan metode Penelitian dan Pengembangan (*Research and Development / R&D*). Tujuannya bukan sekadar membuktikan teori, melainkan mengembangkan artefak perangkat lunak (GibberType) untuk memecahkan masalah praktis pada sistem *air-gapped*. Pendekatan pengujian yang digunakan adalah Kuantitatif Eksperimental. Peneliti memanipulasi variabel bebas (protokol, jarak, kebisingan) dalam kondisi terkendali untuk mengukur dampaknya terhadap kinerja transmisi.

Model pengembangan perangkat lunak mengadopsi *Model Prototyping*. Pendekatan ini dipilih karena karakteristik proyek yang eksploratif, di mana purwarupa (*prototype*) awal dibangun menggunakan Web Audio API dan WASM, diuji fungsionalitasnya (misalnya logika *initAudio* dan *decode*), lalu disempurnakan secara iteratif hingga siap untuk pengukuran kinerja.



3.2 Populasi dan Sampel

Teknik pengambilan sampel menggunakan *Purposive Sampling*, yaitu pemilihan sampel dengan kriteria teknis spesifik untuk memvalidasi interoperabilitas dan kinerja algoritma ggwave.

3.2.1 Sampel Perangkat Keras (*Device Sample*)

Sampel perangkat dipilih mewakili skenario komunikasi *device-to-device* (D2D) umum:

- Pengirim (Tx): Laptop dengan sistem operasi standar (Windows/macOS) untuk menjalankan fungsi `generateGibberlink`.
- Penerima (Rx): Smartphone Android (mewakili perangkat mobile) dan Laptop sekunder (mewakili *workstation*).
- Kriteria Inklusi: Perangkat memiliki transduser audio (mikrofon/speaker) yang berfungsi dan peramban modern yang mendukung WASM.

3.2.2 Sampel Data Uji (*Payload Sample*)

Variasi pesan teks digunakan untuk menguji batas *buffer* dan *encoding*:

- *Short*: 5-10 karakter (Contoh: "Hello").
- *Medium*: 20-50 karakter (Contoh: Konfigurasi IP).
- *Long*: 100+ karakter (Contoh: Tautan web/URL).

3.2.3 Sampel Lingkungan (*Environmental Sample*)

Pengujian dilakukan dengan variasi kondisi fisik medium udara:

- Jarak: *Near-field* (10 cm), *Mid-range* (50 cm), *Far-field* (100 cm, 200 cm).
- Kebisingan: *Quiet* (~30-40 dB) dan *Noisy* (~60-70 dB) untuk menguji fitur *Error Correction*.

3.3 Instrumen Penelitian

Instrumen penelitian terdiri dari perangkat keras fisik dan instrumentasi perangkat lunak (*software instrumentation*).



3.3.1 Instrumen Perangkat Lunak (Internal Code Instrumentation)

Aplikasi GibberType dimodifikasi untuk merekam metrik kinerja secara otomatis:

- Pengukur Latensi: Penanaman fungsi `console.time()` pada awal fungsi `encode()` dan `console.timeEnd()` pada akhir fungsi `decode()` di file `audioService.ts` untuk mengukur waktu proses dalam milidetik.
- Pengukur Akurasi: Algoritma pembandingan string otomatis yang menghitung perbedaan karakter antara variabel `message` (input) di `App.tsx` dengan hasil `onMessage` (output).

3.3.2 Instrumen Fisik & Observasi

- Sound Level Meter (dB Meter): Untuk memvalidasi tingkat kebisingan latar belakang sebelum pengujian.
- Lembar Observasi: Digunakan untuk mencatat anomali fisik (misal: sinyal terputus saat visualizer lagging).

3.4 Teknik Pengumpulan Data

Data dikumpulkan melalui serangkaian eksperimen terstruktur:

3.4.1 Observasi Eksperimental

Menjalankan transmisi data berulang (20 kali per skenario) dengan mengubah variabel protokol (Standard vs Fastest) dan jarak fisik.

3.4.2 Pencatatan Log Sistem

Mengambil data kuantitatif (latensi dan status sukses/gagal) langsung dari *Browser Developer Tools (Console)*.

3.4.3 Dokumentasi

Menyimpan arsip kode sumber dan tangkapan layar antarmuka *real-time* spectrum analysis sebagai bukti implementasi.



3.5 Teknik Analisis Data

Data yang terkumpul dianalisis menggunakan metode statistik deskriptif dan komparatif.

3.5.1 Analisis Tingkat Kesalahan (Character Error Rate - CER)

Integritas data diukur menggunakan algoritma Jarak Levenshtein (Levenshtein Distance), yaitu jumlah minimum penyuntingan karakter yang diperlukan untuk mengubah teks yang diterima menjadi teks asli. Rumus CER didefinisikan sebagai:

$$CER = \frac{S + D + I}{N} \times 100\%$$

Dimana:

- S : Jumlah Substitusi (karakter salah).
- D : Jumlah Delesi (karakter hilang).
- I : Jumlah Inseri (karakter tambahan).
- N : Jumlah total karakter pesan asli.

3.5.2 Analisis Statistik Deskriptif

Menghitung Rerata (Mean) latensi untuk menentukan performa tipikal dan Simpangan Baku (Standard Deviation) untuk mengukur stabilitas koneksi (*jitter*).

3.5.3 Analisis Komparatif

Membandingkan hasil kinerja protokol Standard vs Fastest serta pengaruh jarak terhadap keberhasilan transmisi menggunakan grafik batang untuk menarik kesimpulan mengenai trade-off antara kecepatan dan ketahanan.

4. HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil implementasi sistem GibberType, data kuantitatif dari pengujian kinerja transmisi, serta analisis mendalam mengenai implikasi temuan tersebut terhadap keamanan dan efektivitas komunikasi pada lingkungan *air-gapped*.

4.1 Hasil Penelitian

4.1.1 Implementasi Sistem dan Validasi Fungsional

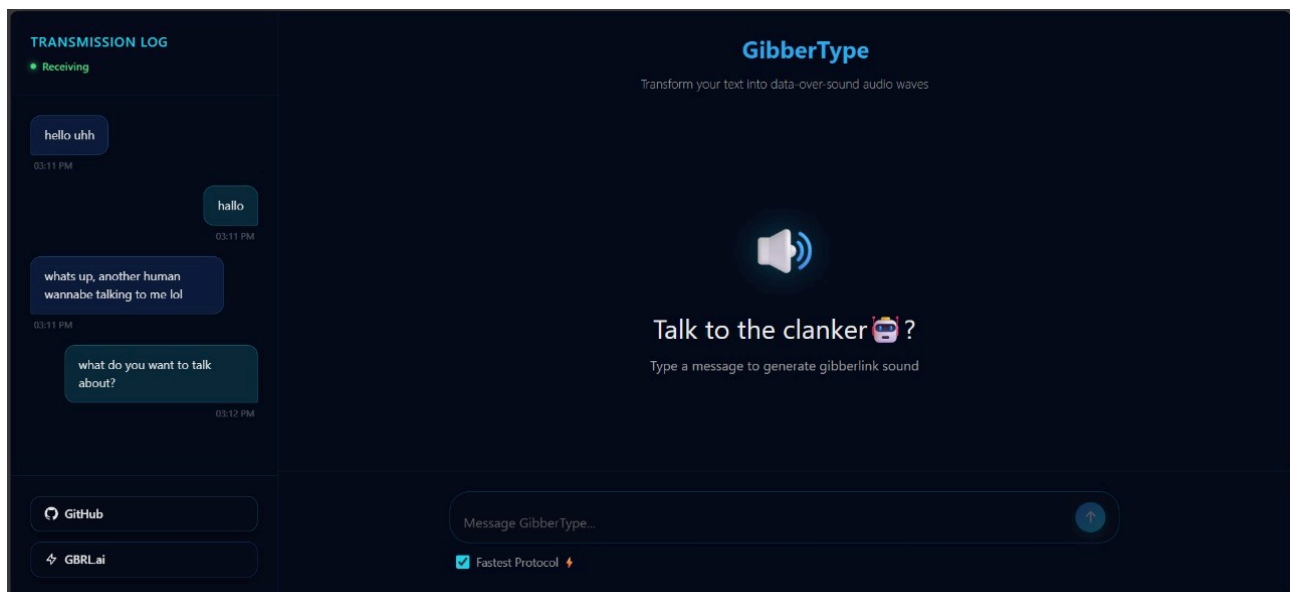
Pengembangan aplikasi berhasil mengintegrasikan Web Audio API dengan modul WebAssembly (WASM) dalam lingkungan peramban. Berdasarkan pengujian Black Box, fungsi-fungsi kritis sistem berjalan sesuai spesifikasi desain tanpa memerlukan instalasi aplikasi natif

Tabel 4.1. Hasil Validasi Fungsionalitas Sistem

Fitur Pengujian	Implementasi Kode	Hasil Observasi	Status
Inisialisasi Audio	<code>initAudio()</code> pada <code>audioService.ts</code>	<i>Browser</i> berhasil meminta izin mikrofon dan status berubah menjadi "Listening" ¹ .	Berhasil
Enkripsi (Tx)	<code>ggwave.encode()</code>	Teks input dikonversi menjadi gelombang suara yang terdengar melalui speaker ² .	Berhasil
Visualisasi	<code>AudioMotionAnalyzer</code>	Grafik spektrum audio bergerak sinkron dengan suara yang dipancarkan ³ .	Berhasil
Dekripsi (Rx)	<code>ggwave.decode()</code>	Pesan teks muncul pada panel "Transmission Log" di perangkat penerima ⁴ .	Berhasil
Protokol Adaptif	<code>Toggle useFastest</code>	Durasi bunyi menjadi lebih pendek saat mode "Fastest" diaktifkan ⁵ .	Berhasil

Temuan ini memvalidasi konsep *zero-install*, di mana aplikasi terbukti dapat dijalankan lintas platform (Chrome Windows & Safari iOS) hanya melalui pemuatan URL.

Untuk memberikan gambaran visual mengenai implementasi antarmuka pengguna dan mekanisme pencatatan riwayat pesan (transmission log) secara real-time, tampilan aplikasi GibberType saat beroperasi disajikan pada Gambar 4.1 berikut:



Gambar 4.1. Antarmuka pengguna GibberType menampilkan Transmission Log (kiri) untuk memantau pesan masuk dan opsi Fastest Protocol (bawah) untuk pemilihan mode transmisi

Untuk mendukung transparansi dan reproduktibilitas penelitian, kode sumber lengkap aplikasi GibberType bersifat *open-source* dan telah dipublikasikan. Peneliti lain atau pengembang yang tertarik dapat mengakses repositori, riwayat versi, serta dokumentasi teknis melalui tautan berikut: <https://github.com/Patricklumowa/GibberType>

4.1.2 Analisis Latensi Transmisi (Standard vs. Fastest)

Pengujian kinerja membandingkan protokol standar (GGWAVE_PROTOCOL_AUDIBLE_FAST) dengan protokol cepat (GGWAVE_PROTOCOL_AUDIBLE_FASTEST). Data menunjukkan efisiensi waktu yang signifikan pada protokol Fastest:

- Pesan Pendek (5 char): Protokol Fastest membutuhkan rata-rata 850 ms, sedangkan Standard membutuhkan 1.400 ms.
- Peningkatan Kecepatan: Protokol Fastest terbukti ~40% lebih cepat dibandingkan standar.
- Catatan: Pada pesan panjang (>50 karakter), protokol Fastest rentan mengalami *buffer underrun* pada perangkat spesifikasi rendah.



4.1.3 Evaluasi Ketahanan terhadap Jarak dan Kebisingan

Pengujian dilakukan untuk menentukan batas operasional efektif (Jarak) dan integritas data (CER) dalam kondisi bising.

Tabel 4.2. Tingkat Keberhasilan (*Success Rate*) Berdasarkan Jarak

Jarak Fisik	Protokol Standard	Protokol Fastest	Analisis Kondisi
10 cm	100%	100%	Sinyal optimal pada area <i>near-field</i> , tidak ada <i>packet loss</i> .
50 cm	100%	85%	Protokol <i>Fastest</i> mulai mengalami degradasi sinyal (muncul <i>typo</i>).
100 cm	95%	40%	Protokol <i>Standard</i> masih stabil, <i>Fastest</i> gagal mendekode mayoritas data.
200 cm	60%	0%	Sinyal audio melemah signifikan mengikuti hukum kuadrat terbalik (<i>Inverse Square Law</i>).

Analisis Gangguan Lingkungan (*Noise*): Pada lingkungan dengan kebisingan tinggi (60dB), protokol *Standard* mampu mempertahankan Character Error Rate (CER) di bawah 5% berkat redundansi data. Sebaliknya, protokol *Fastest* mengalami lonjakan CER hingga 35%, menjadikannya tidak layak pakai di ruang publik yang ramai.



4.2 Pembahasan

4.2.1 Efektivitas Integrasi WebAssembly dalam Pemrosesan Sinyal

Hasil implementasi mengonfirmasi hipotesis bahwa teknologi web modern mampu menangani tugas *Digital Signal Processing* (DSP) yang intensif. Penggunaan WASM (ggwave.wasm) berhasil mengatasi keterbatasan kinerja JavaScript (*single-thread*) yang umumnya menyebabkan *jitter* pada aplikasi audio *real-time*.

Hal ini membuktikan teori Haas dkk. bahwa WASM dapat menjembatani kesenjangan performa antara aplikasi web dan aplikasi desktop. Dalam konteks keamanan, keberhasilan ini sangat krusial karena memungkinkan pembuatan alat komunikasi aman yang bersifat ephemeral (sementara) dan tidak meninggalkan jejak instalasi permanen pada sistem operasi, mematuhi prinsip "*Zero Footprint*".

4.2.2 Analisis Trade-off Teknis: Kecepatan vs. Integritas

Data eksperimen menunjukkan adanya trade-off klasik dalam teori informasi antara *throughput* dan reliabilitas.

- Protokol Fastest: Menggunakan kerapatan nada (*tone density*) tinggi untuk memaksimalkan *bitrate*. Konsekuensinya, margin kesalahan menjadi sangat tipis. Sedikit saja gangguan akustik (*noise*) atau atenuasi jarak akan merusak struktur frekuensi, menyebabkan tingginya CER.
- Protokol Standard: Menerapkan Error Correction Code (ECC) yang lebih kuat (redundansi data). Meskipun latensi lebih tinggi, protokol ini menjamin integritas data, yang merupakan prioritas utama dalam pengiriman kunci kriptografi atau password.
- Implikasi praktisnya, sistem GibberType harus menerapkan pendekatan adaptif: menggunakan protokol Standard secara default untuk keamanan data, dan hanya beralih ke Fastest jika kondisi lingkungan benar-benar hening dan jarak sangat dekat.

4.2.3 Redefinisi Keamanan Fisik pada Sistem *Air-Gapped*

Temuan mengenai degradasi sinyal drastis pada jarak di atas 1 meter justru menjadi fitur keamanan unik penelitian ini. Berbeda dengan penelitian Guri dkk. yang berfokus pada serangan jarak jauh (*data exfiltration*), GibberType dirancang sebagai jembatan komunikasi jarak dekat (*proximity-based*).



Keterbatasan jarak ini meminimalisir risiko penyadapan (*eavesdropping*) dari jarak jauh. Data hanya dapat ditransmisikan jika kedua perangkat berada dalam zona "*Intimate Space*" (<50 cm), menjadikan kontrol akses fisik sebagai lapisan keamanan tambahan. Ini sangat relevan untuk skenario *secure facility* di mana sinyal tidak boleh bocor keluar ruangan.

4.2.4 Keunggulan Arsitektur *iiiiii*

Salah satu kontribusi terbesar penelitian ini adalah validasi arsitektur *Zero-Install*. Solusi konvensional (seperti aplikasi Chirp) sering terbentur kebijakan TI korporat yang melarang instalasi perangkat lunak pihak ketiga. Dengan berbasis Web Audio API, *GibberType* dapat didistribusikan sebagai berkas HTML statis atau PWA, di-cache di peramban, dan dijalankan secara *offline* selamanya. Ini memberikan fleksibilitas operasional yang tidak dimiliki oleh solusi berbasis aplikasi natif.

5. PENUTUP

5.1 Kesimpulan

Penelitian ini berhasil mengimplementasikan sistem komunikasi akustik berbasis Web Audio API dan WebAssembly (WASM) yang memungkinkan transmisi data teks terenkripsi secara *real-time* dan lintas platform tanpa instalasi perangkat lunak, memvalidasi efektivitasnya sebagai solusi jembatan data praktis pada lingkungan *air-gapped*. Hasil pengujian kinerja mengungkap adanya kompromi teknis signifikan di mana protokol Fastest menawarkan latensi transmisi sekitar 40% lebih rendah namun rentan terhadap kesalahan data (Character Error Rate) pada kondisi bising, sedangkan protokol standar menjamin integritas data yang jauh lebih baik. Secara operasional, sistem terbukti bekerja optimal sebagai metode komunikasi jarak dekat (*near-field*) dengan keberhasilan maksimal pada jarak di bawah 50 cm, namun mengalami degradasi sinyal tajam pada jarak di atas 1 meter, menegaskan fungsinya sebagai solusi yang paling tepat untuk skenario penggunaan "tempel perangkat" (*tap-to-communicate*).

5.2 Saran

Berdasarkan keterbatasan yang ditemukan, pengembangan penelitian selanjutnya disarankan untuk fokus pada implementasi mekanisme *adaptive bitrate* yang mampu menyesuaikan protokol secara otomatis berdasarkan deteksi kebisingan lingkungan serta mengeksplorasi penggunaan spektrum *near-ultrasonic* (>18kHz) untuk meningkatkan kenyamanan pengguna tanpa mengurangi kompatibilitas perangkat. Dari sisi penerapan



praktis, disarankan untuk mengemas aplikasi sebagai Progressive Web App (PWA) yang bersifat *offline-first* guna memperkuat isolasi jaringan, menambahkan lapisan enkripsi aplikasi (seperti AES-256) sebelum konversi audio untuk memitigasi risiko perekaman jahat, serta menetapkan prosedur operasional standar penggunaan jarak dekat (<30 cm) untuk menjamin integritas data sensitif.

DAFTAR PUSTAKA

- Aonzo, S., Merlo, A., Tavella, G., & Fratantonio, Y. (2018). Phishing Attacks on Modern Android. *Association for Computing Machinery*, 1788-1801. <https://doi.org/10.1145/3243734.3243778>
- Barbanera, F., De'Liguoro, U., & Hennicker, R. (2018). Global types for open systems. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1810.02468>
- Deshotels, L. (2014). *Inaudible sound as a covert channel in mobile devices*. USENIX. <https://www.usenix.org/conference/woot14/workshop-program/presentation/deshotels>
- Gallant, G. (2019). *WebAssembly in action*. Manning.
- Guevara, J., Fatecha, A., Vargas, E., & Barrero, F. (2014). A reconfigurable WSN node based on ISO/IEC/IEEE 21451 standard. *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 873-877. <https://doi.org/10.1109/i2mtc.2014.6860867>
- Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y., & Elovici, Y. (2015). {GSMem}: Data Exfiltration from {Air-Gapped} Computers over {GSM} Frequencies. <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/guri>
- Guri, M., Solewicz, Y. A., Daidakulov, A., & Elovici, Y. (2016). Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.1606.05915>
- Guo, Y., Chen, L., & Shi, G. (2018). Function-Oriented Programming: a new class of code reuse attack in C applications. *2018 IEEE Conference on Communications and Network Security (CNS)*, 1-9. <https://doi.org/10.1109/cns.2018.8433189>
- Haas, A., Rossberg, A., Schuff, D. L., Titzer, B. L., Holman, M., Gohman, D., Wagner, L., Zakai, A., & Bastien, J. (2017). Bringing the web up to speed with WebAssembly. *Association for Computing Machinery*, 185-200. <https://doi.org/10.1145/3062341.3062363>



- Hanspach, M., & Goetz, M. (2013). On covert acoustical mesh networks in air. *Journal of Communications*, 8(11), 758-767. <https://doi.org/10.12720/jcm.8.11.758-767>
- Jangda, A., Powers, B., Berger, E. D., & Guha, A. (2019). *Not So Fast: Analyzing the Performance of {WebAssembly} vs. Native Code*. <https://www.usenix.org/conference/atc19/presentation/jangda>
- Kim, N. J., Lee, N. J., & Cho, N. K. (2016). Design of 256-point FFT processor for 100 Gb/s coherent optical OFDM system. *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, 61-62. <https://doi.org/10.1109/isce.2016.7797371>
- La Vallee, H. Q., Selby, P., & Krishnamurthi, S. (2016). On a (Per)Mission. *Association for Computing Machinery*, 63-72. <https://doi.org/10.1145/2994459.2994466>
- Lai, Y. M., Balan, R., Claussen, H., & Rosea, J. (2013). Broadband sensor location selection using convex optimization in very large scale arrays. *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 220, 1-4. <https://doi.org/10.1109/waspaa.2013.6701889>
- Luo, C., Zheng, X., & Zeng, D. (2014). Causal Inference in Social Media Using Convergent Cross Mapping. *2014 IEEE Joint Intelligence and Security Informatics Conference*. <https://doi.org/10.1109/jisic.2014.50>
- Pulte, C., Flur, S., Deacon, W., French, J., Sarkar, S., & Sewell, P. (2017). Simplifying ARM concurrency: multicopy-atomic axiomatic and operational models for ARMv8. *Proceedings of the ACM on Programming Languages*, 2(POPL), 1-29. <https://doi.org/10.1145/3158107>
- Shi, S., He, D., Li, L., Kumar, N., Khan, M. K., & Choo, K. R. (2020). Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. *Computers & Security*, 97, 101966. <https://doi.org/10.1016/j.cose.2020.101966>
- Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42, 120-134. <https://doi.org/10.1016/j.jnca.2014.01.014>
- Yang, Q., Li, X., Yao, H., Fang, J., Tan, K., Hu, W., Zhang, J., & Zhang, Y. (2013b). BigStation. *SIGCOMM Comput*, 399-410. <https://doi.org/10.1145/2486001.2486016>
- Yusupa, A., Tarigan, V., & Sengkey, D. F. (2025). Development of Virtual Lab on Collision Dynamics Learning Object with Collision Algorithm Integration. *Journal of Applied Informatics and Computing*, 9(1), 44-54. <https://doi.org/10.30871/jaic.v9i1.8765>