



## Perancangan audio *streaming* menggunakan *wifi* berbasis mikrokontroler atmega 328

Muhammad Satya Annas<sup>1,\*</sup>, Dimas Maulana<sup>2</sup>

<sup>1,2</sup>Sistem Informasi, Universitas Telkom Bandung, Indonesia

\*Correspondence: E-mail: [satya.annas@tutanota.com](mailto:satya.annas@tutanota.com)

ABSTRACT	ARTICLE INFO
<p>Perkembangan teknologi elektronika digital saat ini mengarah pada semakin mengecilnya perangkat keras dengan kemampuan komputasi yang kuat. Begitu juga dengan perangkat audio streaming yang selama ini membutuhkan server PC, telah memungkinkan untuk digantikan dengan mikrokontroler. Tujuan dari penelitian ini adalah untuk menciptakan sistem yang dapat menyiarkan sinyal audio melalui Wi-Fi dari komputer ke set speaker. Dari sejumlah pilihan yang ada, didapatkan bahwa protokol Wi-Fi paling memungkinkan digunakan sebagai metode komunikasi karena perangkat Wi-Fi telah dipasang pada semua laptop modern. Sistem ini memungkinkan pengguna untuk memutar file audio dari komputer ke speaker dalam jangkauan jaringan nirkabel. Dalam penelitian ini bertujuan memanfaatkan penggabungan penggunaan perangkat keras, pemrograman perangkat lunak tingkat rendah, dan protokol standar IEEE 802.11 untuk komunikasi nirkabel (Wi-Fi). Pada penelitian ini digunakan pengkodean data audio oleh encoder sesuai dengan spesifikasi format file WAV. Hasilnya didapatkan bahwa mikrokontroler ATmega328 mampu mengirimkan audio streaming 2 kanal 8 bit dengan sistem yang telah direncanakan.</p>	<p><b>Article History:</b>                      Submitted/Received 27 Dec 2023                      First Revised 24 Feb 2023                      Accepted 13 Mar 2023                      First Available online 28 Mar 2023                      Publication Date 01 Apr 2023</p> <hr/> <p><b>Keyword:</b>                      Audio Streaming,                      Mikrokontroler ATmega328,                      Wi-fi.</p>
<p>© 2023 Universitas Pendidikan Indonesia</p>	

## 1. PENDAHULUAN

Perkembangan teknologi audio digital, khususnya audio *streaming* saat ini telah sampai pada tahapan pemanfaatan perangkat keras mikro yang bisa dibawa kemanapun (*pertable*) (Ashbourn, 2020). Hal ini dimungkinkan dengan keberadaan mikroprosesor dan mikrokontroler yang terus berkembang semakin kuat dalam hal komputasi (Adegbija, et.al. 2017). Sehingga dalam banyak hal, perangkat mikro ini telah dapat menggantikan keberadaan PC atau laptop dengan kemampuan komputasi yang setara.

Konsep awal penelitian ini berasal dari ide transmisi audio nirkabel dari laptop di sebuah ruangan ke *speaker* aktif di ruangan yang lain (Harliantara, 2019). Tujuan dari penelitian adalah agar sinyal audio mencapai jarak yang melebihi kemampuan *bluetooth* saat ini dan tanpa perlu menghubungkan dongle USB eksternal ke komputer. Beberapa penelitian terdahulu menggunakan perangkat non-mikrokontroler dalam melakukan proses audio *streaming* (Lu, et.al 2019). Dari sejumlah pilihan yang ada, didapatkan bahwa protokol *Wi-Fi* paling memungkinkan digunakan sebagai metode komunikasi karena perangkat *Wi-Fi* telah dipasang pada semua laptop modern. Dari beberapa produk komersial sejenis yang ada, ditemukan bahwa jarang sekali produk yang menawarkan layanan *streaming* audio melalui *Wi-Fi*. Penelitian lebih lanjut dilakukan untuk menentukan mikrokontroler yang paling sesuai untuk dihubungkan dengan modul *Wifi-Shield* (Rizal-Alfariski, et.al. 2022). Penelitian ini menemukan papan mikrokontroler *Arduino* (*Arduino Board*) yang mengandung mikrokontroler Atmega328P paling sesuai untuk antarmuka dengan *Wifi Shield* untuk koneksi tersertifikasi melalui jaringan 802.11b/g

Pada penelitian ini digunakan pengkodean data audio oleh *encoder* sesuai dengan spesifikasi format file WAV. WAV bersama MP3 dan Ogg forbis adalah format yang paling sering digunakan pada audio digital.

Berikut merupakan tinjauan pustaka yang telah dilakukan:

### 1.1. *Wi-fi* Standar IEEE 802.11

Standar protokol IEEE 802.11 adalah bagian dari protokol LAN IEEE 802, yang menerapkan seperangkat MAC dan protokol lapisan fisik (PHY) pada jaringan area lokal nirkabel (WLAN) dalam berbagai frekuensi, yaitu 2,4, 5, dan 60 GHz.

IEEE 802.11 adalah standar jaringan komputer nirkabel yang paling banyak digunakan di dunia (Tangkowit, et.al. 2021). Digunakan di sebagian besar jaringan rumah dan kantor untuk memungkinkan laptop, *printer*, dan *smartphone* saling berkomunikasi dan mengakses Internet tanpa menghubungkan kabel. Protokol ini dibuat dan dikelola oleh *Institute of Electrical and Electronics Engineers* (IEEE) LAN / MAN *Standards Committee* (IEEE 802). Versi dasar dari standar ini dirilis pada tahun 1997, dan telah mengalami banyak modifikasi (Abusubaih, et.al. 2013).

Protokol 802.11b dan 802.11g menggunakan pita 2,4 GHz ISM, yang beroperasi di Amerika Serikat berdasarkan Peraturan dan Regulasi Komisi Komunikasi Federal AS; 802.11n juga dapat menggunakan band itu. Karena pilihan pita frekuensi ini, peralatan 802.11b / g / n kadang-kadang dapat mengalami gangguan pada pita 2,4 GHz dari oven *microwave*, telepon nirkabel, dan perangkat Bluetooth dll. Protokol 802.11b dan 802.11g mengendalikan gangguan dan kerentanannya terhadap gangguan dengan menggunakan metode penyebaran *direct-sequence spread spectrum* (DSSS) dan *orthogonal frequency-division multiplexing* (OFDM) (Afdhal, A., & Elizar, E. 2014). Tabel 1 menunjukkan perbandingan protokol IEEE 802.11b/g/n.

Tabel 1. Perbandingan beberapa Standar Protokol IEEE 802

Standar	Frekuensi	Bandwidth	Jangkauan
IEEE 802.11b	2.4 GHz	22 MHz	35 m
IEEE 802.11g	2.4 GHz	5/10/20 MHz	38 m
IEEE 802.11n	2.4 & 5 GHz	50 MHz	70 m

## 1.2. Audio Streaming

*Streaming* audio adalah audio yang secara konstan diterima oleh dan disajikan kepada *user* saat dikirimkan oleh penyedia. Kata kerja "*streaming*" mengacu pada proses pengiriman atau perolehan media dengan cara ini. Sehingga istilah ini merujuk pada metode pengiriman, dan merupakan alternatif untuk mengunduh file, suatu proses di mana *user* mendapatkan seluruh file untuk konten sebelum menonton atau mendengarkannya (Haritha & Reddy, 2014).

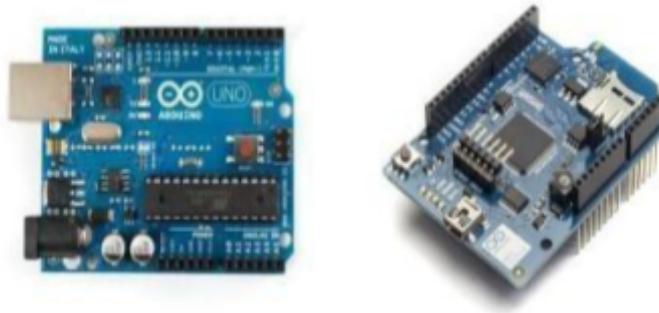
Sedangkan *live streaming* atau *real-time streaming* adalah pengiriman konten secara *real-time* seperti halnya siaran siaran langsung konten audio melalui gelombang udara pada siaran radio (Work, 2018). *Streaming* internet langsung memerlukan bentuk media sumber (antarmuka audio, perangkat lunak audio *player*), enkoder untuk mendigitalkan konten, server, dan jaringan pengiriman konten untuk mendistribusikan dan mengirimkan konten (Parmawati, 2015). Ada tantangan dengan *streaming* audio di jaringan. Jika pengguna tidak memiliki *bandwidth* yang cukup dalam koneksi jaringan, mungkin mengalami penghentian, keterlambatan, atau *buffering* konten yang lambat. Beberapa pengguna mungkin tidak dapat melakukan *streaming* audio tertentu karena tidak memiliki perangkat atau sistem perangkat lunak yang kompatibel. Teknologi audio digunakan untuk mendistribusikan *streaming* biasanya menggunakan *codec audio lossy*. Format audio yang termasuk *streaming* adalah MP3, Ogg Vorbis, *Windows Media Audio* (WAV), RealAudio, dan HE-AAC (atau aacPlus). Data audio secara terus menerus ditransmisikan secara serial melalui jaringan lokal atau internet dalam paket TCP atau UDP, kemudian dipasang kembali di penerima dan diputar satu atau dua detik kemudian. Penundaan ini disebut *lag*, dan sering terjadi pada beberapa tahap penyiaran audio digital.

## 1.3. Arduino UNO Board dan Modul Wi-fi

UNO *Arduino* adalah *board development* mikrokontroler *open-source* berbasis ATmega328P dan dikembangkan oleh Arduino.cc. *Board* ini dilengkapi dengan sekelompok pin *input/output* (I/O) digital dan analog yang dapat dihubungkan ke berbagai *board* ekspansi (*Shield*) dan rangkaian elektronik lainnya. *Arduino* UNO memiliki 13 pin Digital, 6 pin Analog, dan dapat diprogram dengan *Arduino IDE* (*Integrated Development Environment*)

melalui kabel USB tipe B. Sumber daya diperoleh dari kabel USB atau sumber daya eksternal antara 7 dan 12 volt (Subagja, 2019).

Desain referensi perangkat keras didistribusikan di bawah lisensi *Creative Commons Attribution Share-Alike 2.5* dan tersedia di situs web *Arduino*. File tata letak dan produksi untuk beberapa versi perangkat keras juga tersedia untuk publik. "Uno" berarti satu dalam bahasa Italia dan dipilih untuk menandai rilis *Arduino Software (IDE) 1.0*. Mikrokontroler ATmega328 pada *Arduino Uno* telah diprogram sebelumnya dengan *bootloader* yang memungkinkan mengunggah kode baru ke dalamnya tanpa menggunakan *programmer* perangkat keras eksternal. Bentuk fisik *Arduino UNO* tampak pada **Gambar 1**.



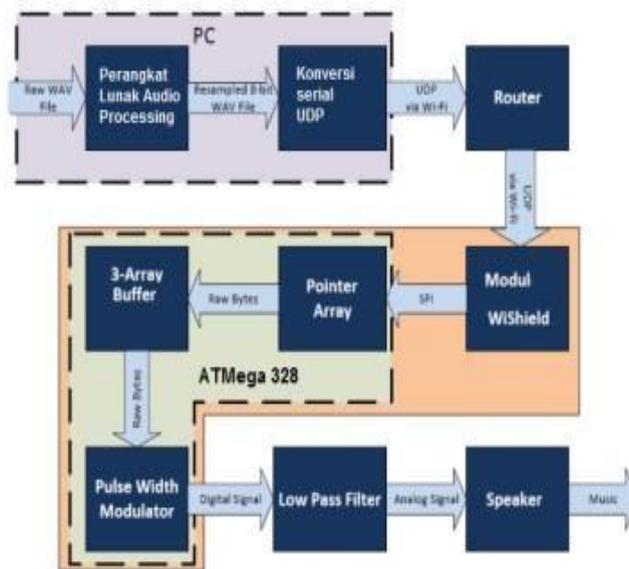
**Gambar 1.** *Arduino UNO dan Wifi Shield v.1*

Sedangkan *Wifi Shield* adalah perangkat tambahan untuk *board Arduino* yang menggunakan modul *wifi RN171* untuk memberikan fungsi serial *Ethernet* pada *Arduino*. Hanya diperlukan dua pin untuk menghubungkan perangkat Anda ke jaringan nirkabel 802.11b / g dengan *Shield* ini. Dengan dukungan untuk protokol komunikasi TCP, UDP, dan FTP, *Wifi Shield* ini dapat memenuhi kebutuhan koneksi jaringan nirkabel. Karena didesain untuk dipasang tepat pada konektor pin *Arduino*, maka seluruh fungsi pin *Arduino* juga akan muncul pada konektor pin *Wifi Shield*. Agar dapat memanfaatkan fitur koneksi nirkabel pada *Wifi Shield*, maka proses pemrograman *Arduino* membutuhkan pustaka yang mendukung *shield board* ini (Subagja, 2019).

## 2. METODE

Streaming audio melalui *Wi-Fi* menggunakan ATmega328 terdiri dari atas beberapa tahapan untuk proses transmisi audio yang baik (Osanaïye, et.al 2022). Tahapan-tahapan ini dikategorikan sebagai berikut:

1. Antarmuka Perangkat Keras
2. Pemrograman
3. *Debugging*



**Gambar 2.** Diagram sistem *streaming* audio via *Wi-fi* menggunakan ATmega328

Sistem yang diusulkan pada **Gambar 2**, dibuat di mana audio disampling dengan frekuensi 22 KHz dari komputer *host*. Audio WAV diproses menjadi paket UDP dari 325 byte dan diteruskan ke mikrokontroler ATmega 328 melalui *router* dan modul *WiShield*. Paket-paket disimpan dalam tiga siklus *buffer array*. Mikrokontroler mengeluarkan *Pulse Width Modulation* pada paket yang diterima. *Low pass filter* digunakan untuk melemahkan frekuensi tinggi yang tidak diinginkan sehingga hasilnya hampir seperti sinyal audio. *Arduino* terhubung dengan *speaker* dari mana kita mendapatkan hasil akhir suara.

### 2.1. Hardware Interfacing

Perangkat keras utama pada penelitian adalah *board development* yang disebut *Arduino UNO* dan *Wifi Shield*. Dalam *board Arduino UNO* sudah terpasang mikrokontroler ATmega328p yang dihubungkan dengan modul *Wi-Fi* melalui koneksi *SPI*. Selain itu, pada *board* juga memiliki beberapa LED untuk indikator daya dan konektivitas *Wi-Fi*. ATmega328p berisi 2 KB RAM. 1 KB digunakan dalam *handshaking* dengan *WiShield*, sehingga menyisakan 1 KB untuk melakukan *buffering* yang diperlukan untuk *streaming*.

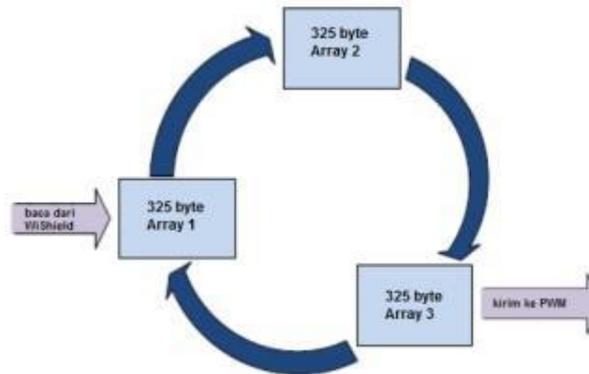
Ukuran *buffer* yang terbatas inilah yang pada akhirnya membatasi kinerja sistem. *WiShield* terhubung ke jaringan nirkabel seperti perangkat lain, sehingga membutuhkan alamat IP statis. Untuk dapat memprogram *Arduino UNO* dibutuhkan koneksi USB dan *driver* yang sesuai.

### 2.2. Pemrograman

Yang harus dilakukan pada tahap awal adalah memprogram *WiShield* untuk terhubung ke jaringan. Parameter jaringan yang diperlukan dari *router* seperti alamat IP, *gateway default*, dan *subnet mask* harus dikodekan ke dalam memori program mikrokontroler. Program *handshaking* sederhana digunakan untuk menentukan ukuran paket maksimum yang dapat diterima oleh *WiShield* yaitu sekitar 325 byte. Untuk aliran kontinu, siklus *buffer* dibuat pada mikrokontroler. Oleh karena dibutuhkan 1 KB RAM untuk tiga siklus *buffer array*, dengan masing-masing *array* terdiri dari tiga 325 byte. *Array* ini dideklarasikan sebagai *array char volatile*, karena akan dibaca dalam *ISR* dan harus berukuran satu *byte*.

Selanjutnya fungsi ISR dibuat dan dipakai sekali pada setiap sampel, atau 1/22 kHz atau sekitar 45 uS. Dalam fungsi ini, satu *byte* dari salah satu dari tiga *array* dibaca dan ditempatkan ke dalam *register* OCR2B. *Register* ini bertanggung jawab untuk mengatur siklus tugas *timer2* PWM. Karena PWM ini berjalan pada kira-kira 62 kHz, setelah sinyal ini disaring menjadi rendah, maka sinyal ini menjadi hampir seperti sinyal audio.

Sebuah variabel yang bertanggung jawab untuk melacak larik mana yang harus dibaca dan ditulis. Setiap 325 *byte* dari ISR, variabel ini diperbarui ke nilai baru untuk secara efektif menukar fungsi *array*. Karena informasi telah terkuras dari salah satu *array*, *array* berikutnya dalam urutan menggantikannya dan *array* kosong diisi dengan data baru. Gambar 3 membantu untuk mengilustrasikan proses ini secara grafis:



Gambar 3. Proses siklus pengisian *array buffer* untuk pada audio *streaming*

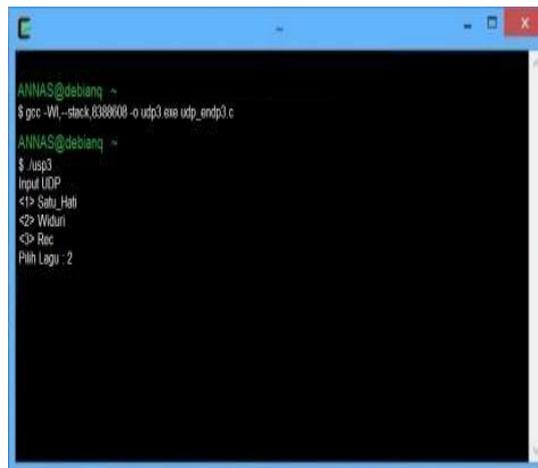
Modulasi lebar pulsa (PWM) bekerja dengan menyandikan nilai analog ke dalam gelombang persegi digital dengan mengubah *duty cycle*. ATmega328p menghasilkan musik dengan menggunakan teknik ini. Frekuensi analog yang lebih rendah dikodekan dalam frekuensi digital yang jauh lebih tinggi dan kemudian dikirim melalui *low pass* filter untuk mendapatkan kembali sinyal yang diinginkan.

### 2.3. Debugging

Bagian penting selanjutnya dari penelitian ini adalah *debugging* karena cukup memakan waktu dan sulit untuk menentukan serta memperbaiki masalah di pemancar dan penerima akhir. Proses ini membutuhkan program yang disebut *Wireshark* yang digunakan untuk menemukan *bug*. Pengiriman paket data yang drop dan hilangnya koneksi akan diketahui pada *Wireshark*. Oleh karena itu dibutuhkan sebuah *script* sederhana untuk menjaga koneksi tetap tersambung.

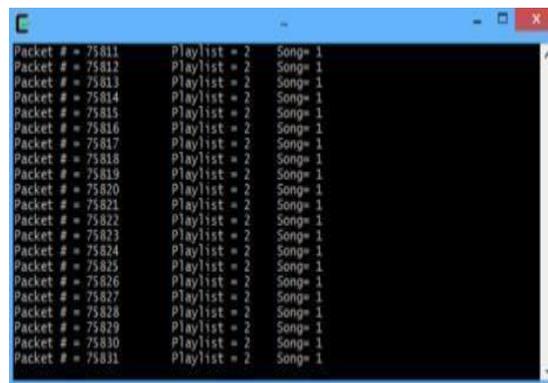
## 3. HASIL DAN PEMBAHASAN

Dalam **Gambar 4** menunjukkan *window Cygwin Command* hasil menu memilih lagu untuk memutar audio yang diinginkan. Audio ini dikirimkan melalui jaringan *wifi* menggunakan protokol UDP.



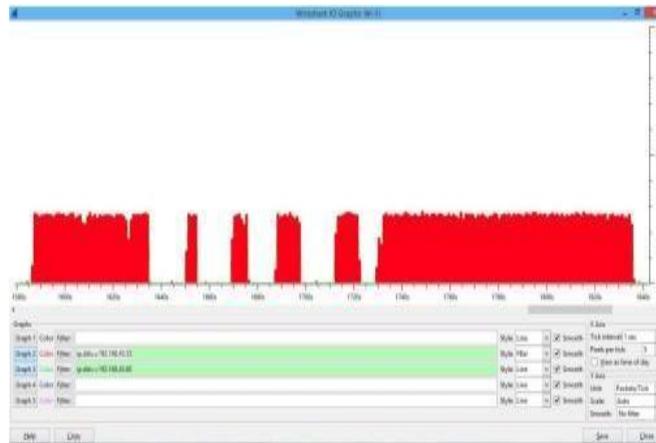
**Gambar 4.** Window Cygwin untuk pengiriman audio dengan UDP

Pada **Gambar 5** menunjukkan bahwa setelah pemilihan lagu, akan tampil ukuran lagu dan tiap transmisi mulai menunjukkan jumlah *real-time* paket yang dikirim dalam jaringan. Melalui proses ini dapat dilihat jika ada paket yang hilang dalam transmisi.

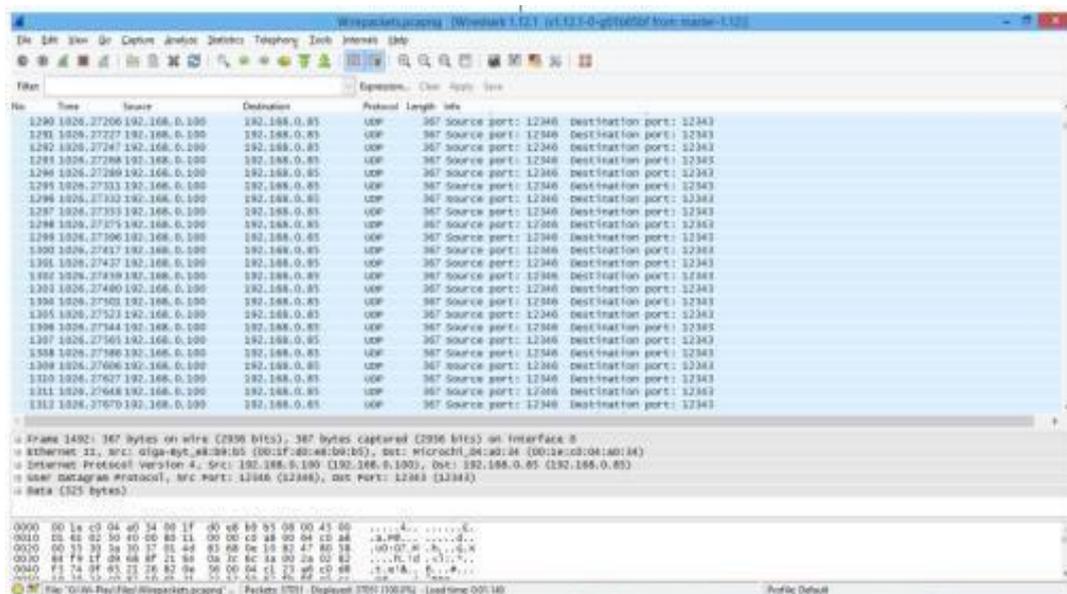


**Gambar 5.** Window Cygwin untuk monitoring pengiriman paket data

Pada **Gambar 6**, grafik merah menunjukkan transmisi data secara *real time* dari ujung komputer. Sumbu x menentukan waktu untuk transmisi audio tertentu. Sedangkan pada **Gambar 7** menunjukkan alamat Sumber, alamat tujuan, Panjang Protokol dan Info Sumber dan Port Tujuan. Gambar ini juga menampilkan sinyal permintaan dan balasan pada log beserta artinya sebagaimana ditentukan oleh protokol jaringan. *Wireshark* menangkap setiap paket yang dikirim menuju atau dari sistem jaringan. Jika yang dimonitor adalah antarmuka nirkabel, maka akan dapat dilihat paket lain di jaringan tersebut. Secara default, hijau adalah trafik TCP, biru gelap adalah lalu lintas DNS, biru muda adalah lalu lintas UDP, dan hitam mengidentifikasi paket TCP dengan masalah.



Gambar 6. Window Wireshark grafik transmisi



Gambar 7. Window Wireshark monitoring paket data

Sedangkan pada **Gambar 7** menunjukkan alamat Sumber, alamat tujuan, Panjang Protokol dan Info Sumber dan *Port Tujuan*. Gambar ini juga menampilkan sinyal permintaan dan balasan pada log beserta artinya sebagaimana ditentukan oleh protokol jaringan. *Wireshark* menangkap setiap paket yang dikirim menuju atau dari sistem jaringan. Jika yang dimonitor adalah antarmuka nirkabel, maka akan dapat dilihat paket lain di jaringan tersebut. Secara *default*, hijau adalah trafik TCP, biru gelap adalah lalu lintas DNS, biru muda adalah lalu lintas UDP, dan hitam mengidentifikasi paket TCP dengan masalah.

#### 4. KESIMPULAN

Hasil percobaan pada penelitian ini berhasil mentransmisikan audio dengan 2 kanal (Stereo) 8-bit dari komputer pada *sample rate* 22,05 KHz. Kualitas audio terdegradasi dari 16-bit stereo WAV menjadi 8-bit Stereo. Meskipun audio berhasil ditransmisikan, ada beberapa masalah yang perlu diperbaiki pada penelitian selanjutnya, yaitu bahwa sesekali terjadi putus koneksi antara pemancar dan penerima untuk rentang waktu kecil sehingga suara yang menghasilkan lagu juga terputus. Hal ini sebaiknya diatasi dengan memastikan jaringan yang stabil dengan perangkat yang memadai.

## 5. CATATAN PENELITIAN

Peneliti dengan tegas menekankan integritas akademik dan komitmen terhadap etika penelitian dan menghindari pelanggaran serius terhadap integritas ilmiah, dan penelitian ini akan dilakukan dengan mematuhi standar tertinggi dalam mencegah praktik plagiarisme. Peneliti bertujuan untuk menjamin bahwa hasil penelitian ini adalah produk original dan bahwa segala bentuk kontribusi dari pihak lain akan diidentifikasi dengan jelas dan diberikan atribusi yang tepat.

## 6. REFERENSI

- Adegbija, T., Rogacs, A., Patel, C., & Gordon-Ross, A. (2017). Microprocessor optimizations for the internet of things: A survey. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1), 7-20.
- Afdhal, A., & Elizar, E. (2014). Ieee 802.11 ac sebagai standar pertama untuk gigabit wireless lan. *Jurnal Rekayasa Elektrika*, 11(1), 36-44.
- Ashbourn, J. (2020). Audio technology, music, and media. *Cham, Switzerland: Springer Nature*. doi, 10, 978-3.
- Harliantara, H. (2019). Website pada industri penyiaran radio di indonesia: live streaming dan podcasting. *Jurnal Studi Komunikasi*, 3(1), 82-100.
- Haritha, S., & REDDY, R. A. (2014). Design and implementation of efficient audio streaming system using raspberry pi. *Int. J. Emerg. Technol. Comput. Sci. Electron. IJETCSE*, 12(1) 33-36.
- Lu, R., Duan, Z., & Zhang, C. (2019). Audio–visual deep clustering for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11), 1697-1712.
- Osanaiye, O. A., Mannan, T., & Aina, F. (2022). An IoT-based soil moisture monitor. *African Journal of Science, Technology, Innovation and Development*, 14(7), 1908-1915.
- Parmawati, P. T. (2015). Pengembangan sistem informasi layanan audio visual perpustakaan universitas pendidikan ganesha berbasis video streaming. *JST (Jurnal Sains Dan Teknologi)*, 4(1), 553-565.
- Rizal-Alfariski, M., Dhandi, M., & Kiswantono, A. (2022) Automatic Transfer Switch (ATS) Using Arduino Uno, IoT-Based Relay and Monitoring. *Jurnal Sistem Telekomunikasi, Elektronika, Sistem Kontrol, Power Sistem dan Komputer*, 2(1), 1-8.
- Subagja, R. E. (2019). Pembuatan trainer on/off lampu menggunakan sms berbasis mikrokontroler arduino uno atmega328. *Jurnal Informatika dan Komputer (JIK)*, 10(1), 27-34.
- Tangkowit, A. E., Palilingan, V. R., & Liando, O. E. S. (2021). Analisis dan perancangan jaringan komputer di sekolah menengah pertama. *Eduatik: Jurnal Pendidikan Teknologi Informasi dan Komunikasi*, 1(1), 69-82.

Work, R. S. (2018). Analisis QOS (Quality Of Service) pengukuran delay, jitter, packet lost dan throughput untuk mendapatkan kualitas kerja radio streaming yang baik. *Jurnal Teknologi Informasi Dan Komunikasi*, 7(2), 98-105.