

**IMPLEMENTASI KOMUNIKASI MIDI MENGGUNAKAN UDP
BROADCAST**

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI REKASAYA KOMPUTER

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



**FAHRIZAL HARI UTAMA
NIM. 145060301111002**

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2018

LEMBAR PENGESAHAN
IMPLEMENTASI KOMUNIKASI MIDI MENGGUNAKAN UDP *BROADCAST*
SKRIPSI

TEKNIK ELEKTRO KONSENTRASI REKAYASA KOMPUTER

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



FAHRIZAL HARI UTAMA
NIM. 145060301111002

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing

Pada tanggal 17 Desember 2018

Telah diperiksa dan disetujui oleh

Dosen Pembimbing 1

Dosen Pembimbing 2


Waru Djuriatno, S.T, M.T
NIP. 19690725 199702 1 001


A. Abdul Razak, S.T., M.T., M.Eng., Ph.D
NIK. 201201 850716 1 001

Mengetahui,
Ketua Jurusan Teknik Elektro

Hadi Suyitno, S.T, M.T., Ph.D, IPM.
NIP. 19730520 200801 1 013



JUDUL SKRIPSI :

IMPLEMENTASI KOMUNIKASI MIDI MENGGUNAKAN UDP *BROADCAST*

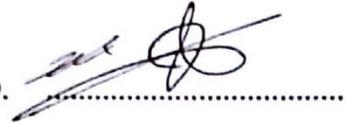
Nama Mahasiswa : Fahrizal Hari Utama
NIM : 145060301111002
Program Studi : Teknik Elektro
Konsentrasi : Rekayasa Komputer

KOMISI PEMBIMBING :

Ketua : Waru Djurianto, S.T., M.T.



Anggota : A. Abdul Razak, S.T., M.T., M.Eng., Ph.D.



TIM DOSEN PENGUJI :

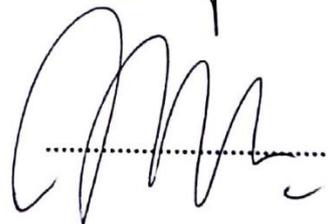
Dosen Penguji 1 : Dr. Ir. M. Aswin, M.T.



Dosen Penguji 2 : Raden Arief Setiawan, S.T., M.T.



Dosen Penguji 3 : Dr. Ir. Bambang Siswojo, M.T.



Tanggal Ujian : 14 Desember 2018

SK Penguji : 2819/UN10.F07/SK/2018



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan, dan masalah ilmiah yang diteliti dan diulas di dalam naskah skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 17 Desember 2018

Mahasiswa,



Fahrizal Hari Utama

NIM 145060301111002

- ၆ ဝဋ် ဂီ ငိပင်. မိလလု အင လေင် ၁-

...: *Maju ke Depan, Selalu Ada Jalan* ...:



တီအိင် အုပင် တီဝိယ ဂမိး ဂိပင်:

အယဟင်င ငင် အိဗင်င တီဝိင်င

ပက မဟဗတီ

..o..

Teriring Ucapan Terima Kasih Kepada:

Ayahanda dan Ibunda tercinta

Para Sahabat

---o---



PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada kehadiran Allah *Subhanahu Wa Ta'ala*, karena atas segala petunjuk, rahmat dan nikmat-Nya lah skripsi ini dapat diselesaikan. Sholawat serta salam tidak lupa penulis aturkan kepada junjungan Rasullullah Muhammad *Shollallahu'alaihi Wa Salam* semoga kelak mendapatkan syafaat beliau di *yaumul qiyamah*.

Skripsi berjudul “Implementasi Komunikasi MIDI Menggunakan UDP *Broadcast*” ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan dan doa berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih yang tak terhingga kepada:

1. Bapak Hadi Suyono, S.T, M.T., Ph.D., IPM dan Ibu Ir. Nurussa'adah, M.T., selaku Ketua dan Sekretaris Jurusan Teknik Elektro Universitas Brawijaya yang selalu memberi semangat dan motivasi untuk cepat menyelesaikan skripsi.
2. Bapak Ali Mustofa, S.T, M.T., selaku Kepala Program Studi Teknik Elektro Universitas Brawijaya.
3. Bapak Adharul Muttaqin, S.T., M.T., selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya yang juga meminjamkan alat bantu dalam penyelesaian skripsi ini.
4. Bapak Waru Djuriatno, S.T., M.T. dan Bapak Angger Abdul Razak, S.T., M.T., M.Eng., Ph.D., selaku dosen pembimbing skripsi, di tengah kesibukan beliau selalu memberikan waktu untuk diskusi dengan tulus dalam memberikan arahan, saran, nasihat, dan masukan yang sungguh berharga untuk membantu penulis dalam menyelesaikan skripsi.
5. Para Dosen Pengajar Program Studi Teknik Elektro Universitas Brawijaya, yang tidak dapat penulis sebutkan satu per satu yang telah memberikan bekal ilmu pada penulis dalam menyelesaikan studi.
6. Kedua orang tua yaitu Bapak Wahyudi, Ibu Nanik Wahyu Suryanti, kakak Mahendra, dan keluarga besar penulis yang selalu memberikan doa, nasihat, motivasi, dan dukungan untuk menyelesaikan studi di Teknik Elektro dari awal hingga akhir studi.
7. Himpunan dan Keluarga Besar Mahasiswa Teknik Elektro, khususnya DIODA 14 yang telah memberikan 4 tahun lebih pengalaman yang luar biasa.
8. Laboran Laboratorium Infokom, kakak-kakak, teman-teman dan adik-adik asisten Laboratorium Infokom, mas Rakhmad Romadhoni, S.T., mas Wekatama, mas Risto, mas Alin, mas Zuhaj, mas Ari, mas Satria, Farhan, Fuad, Faris dan adik-adik 2015 serta 2016 atas segala dukungan dalam pengerjaan skripsi.
9. Keluarga besar RiSTIE UB yang telah menyediakan tempat beristirahat, berbagi, dan belajar, di lingkungan kampus siang malam. Terima Kasih.
10. Teman-teman satu kontrakan 338B yang sudah memberikan pengalaman 3 tahun bersama terutama Rahmat dengan 4 tahun berseberangan kamar. Terima kasih banyak.
11. Teman-teman seru-seruan di grup Line “Dosa. 🐱”, Farhan, Fuad, Fondra, Ghiyas, Ion, Weeb Overseer dan Mas Bagus yang selalu memberikan semangat dan motivasi ketika

- sedang *down* dan malas sampai saat ini, terutama “Gussygoose” yang memberi hiburan dan seni secara gratis.
12. Teman-teman se-PKM, Ika, Mifta, Nimas, Reynaldi, mas Rynaldi, dan Sidiq, terima kasih karena memberikan pengalaman dan gambaran luas ilmu pengetahuan di luar dan di dalam kampus.
 13. Teman seperantauan di Malang, Ikatan Mahasiswa Kendal Malang yang saling mengingatkan satu sama lain dalam berbagai hal. Terima kasih telah menemani selama di Malang.

Sekiranya Allah SWT membalas kebaikan semua pihak yang turut membantu skripsi ini terselesaikan. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari sempurna, namun semoga skripsi ini dapat memberikan manfaat bagi kita semua. Amiin, Terima kasih.

Malang, 4 Desember 2018

Penulis



RINGKASAN

Fahrizal Hari Utama, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2018, Implementasi Komunikasi MIDI Menggunakan UDP *Broadcast*, Dosen Pembimbing: Waru Djuriatno dan Angger Abdul Razak.

Secara praktik, Jumlah perangkat yang dapat terhubung menggunakan *daisy chain* MIDI THRU terbatas. Hal itu disebabkan lokasi alat musik yang tersebar dalam ruangan atau berada di ruangan/lantai yang berbeda. Tujuan dari penelitian ini adalah untuk mengimplementasikan komunikasi MIDI melalui jalur UDP Broadcast menggunakan Wi-Fi. Proses pengiriman pesan MIDI menjadi paket UDP adalah dengan membungkus satu/banyak pesan MIDI menjadi paket UDP dengan alamat broadcast jaringan. Paket yang sampai pada tujuan kemudian dibuka dan dikembalikan menjadi pesan MIDI. Modul komunikasi yang digunakan pada pengirim dan penerima adalah ESP32 *Dev Board* dengan rangkaian pengondisi sinyal MIDI. Program yang digunakan dalam penelitian ini adalah *Arduino Framework* dengan pustaka ESP32 dan UDP Asinkron. Dalam jurnal ini hasil penelitian akan menunjukkan nilai jeda yang muncul akibat proses pembungkusan dan pengiriman pesan MIDI.

Kata kunci— MIDI, UDP, Wi-Fi, ESP32, Jeda.

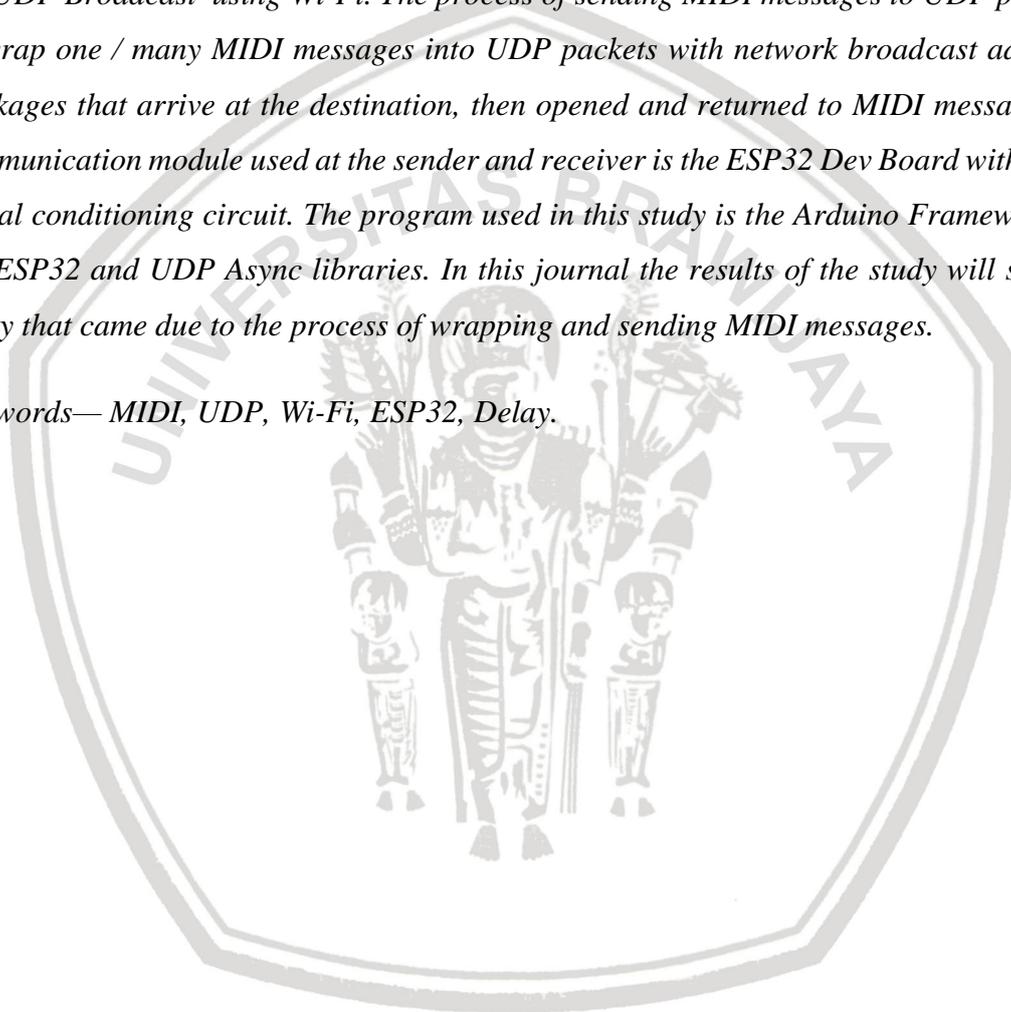


SUMMARY

Fahrizal Hari Utama, *Department of Electrical Engineering, Faculty of Engineering, Brawijaya University, December 2018, Implementation MIDI Communication Using UDP Broadcast, Academic Supervisor: Waru Djuriatno and Angger Abdul Razak.*

In practice, the number of devices that can be connected using daisy chains on THRU MIDI is limited. This is due to the location of musical instruments spreaded in the room or in different rooms / floors. The purpose of this research is to implement MIDI communication via UDP Broadcast using Wi-Fi. The process of sending MIDI messages to UDP packets is to wrap one / many MIDI messages into UDP packets with network broadcast addresses. Packages that arrive at the destination, then opened and returned to MIDI messages. The communication module used at the sender and receiver is the ESP32 Dev Board with a MIDI signal conditioning circuit. The program used in this study is the Arduino Framework with the ESP32 and UDP Async libraries. In this journal the results of the study will show the delay that came due to the process of wrapping and sending MIDI messages.

Keywords— MIDI, UDP, Wi-Fi, ESP32, Delay.





DAFTAR ISI

PENGANTAR	i
RINGKASAN	iii
SUMMARY	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 MIDI.....	5
2.1.1 Pesan MIDI.....	6
2.1.2 Pocket MIDI.....	7
2.1.3 LoopMIDI.....	8
2.1.4 USB-MIDI	9
2.2 UDP	9
2.3 ESP32 <i>Dev Board</i>	10
2.4 Wi-Fi	11
2.5 Visual Studio Code.....	12
2.5.1 Ekstensi Platform IO.....	12
2.6 Node-RED.....	12
2.7 <i>Level Shifter</i>	13
BAB III METODE PENELITIAN.....	15
3.1 Metode Penelitian.....	15
3.2 Studi Literatur.....	15
3.3 Perancangan dan Penentuan Spesifikasi Sistem.....	16
3.3.1 Spesifikasi Bagian Pengirim	16
3.3.2 Spesifikasi Bagian Penerima.....	19
3.3.3 Pengaturan <i>Access Point</i>	20



3.4	Langkah Kerja Sistem	21
3.4.1	Instalasi dan Konfigurasi Program	24
3.4.2	Program Pada Modul Pengirim	25
3.4.3	Program Pada Modul Penerima	25
3.5	Pengujian dan Analisis Sistem	25
3.5.1	Pengujian Sistem Membaca Data MIDI	25
3.5.2	Pengujian Komunikasi UDP	26
3.5.3	Pengujian Waktu Tunda Sistem	26
3.6	Analisa Hasil dan Kesimpulan	26
BAB IV HASIL DAN PEMBAHASAN.....		27
4.1	Pengujian Sistem Membaca Data MIDI	27
4.1.1	Prosedur Percobaan	27
4.1.2	Hasil Pengujian dan Analisis	28
4.2	Pengujian Komunikasi UDP	29
4.2.1	Prosedur Percobaan	30
4.2.2	Hasil Pengujian dan Analisis	31
4.3	Pengujian Waktu Tunda Sistem	31
4.3.1	Prosedur Percobaan	32
4.3.2	Hasil Pengujian dan Analisis	32
BAB V KESIMPULAN DAN SARAN		37
5.1	Kesimpulan	37
5.2	Saran.....	37
DAFTAR PUSTAKA		39
LAMPIRAN.....		41
	Lampiran 1 Kode Sumber Perangkat Pengirim “midi_type.h”	42
	Lampiran 2 Kode Sumber Perangkat Pengirim “main.cpp”	43
	Lampiran 3 Kode Sumber Perangkat Penerima “main.cpp”	47
	Lampiran 4 Kode Sumber Pengujian 1 “main.cpp”	49
	Lampiran 5 Kode Sumber Pengujian 2 Pengirim "main.cpp"	53
	Lampiran 6 Kode Sumber Pengujian 2 Penerima “main.cpp”	55
	Lampiran 7 Kode Sumber Arduino Mega 2560 “DMS_MEGA_2560.ino”	57
	Lampiran 8 Pengujian <i>drop</i> Paket Komunikasi UDP ESP.....	58



DAFTAR GAMBAR

Gambar 2.1	Daisy Chain MIDI.....	5
Gambar 2.2	Rangkaian Elektronik MIDI.....	6
Gambar 2.3	Bagan Pembagian Pesan MIDI.....	7
Gambar 2.4	Struktur Pesan MIDI.....	7
Gambar 2.5	Byte Pesan MIDI.....	7
Gambar 2.6	Antarmuka MIDI-OX.....	8
Gambar 2.7	Antarmuka loopMIDI.....	8
Gambar 2.8	USB-MIDI.....	9
Gambar 2.9	Header UDP.....	9
Gambar 2.10	Header UDP IPv4.....	10
Gambar 2.11	Papan ESP32 <i>Dev Board</i>	11
Gambar 2.12	Antarmuka Visual Studio Code.....	12
Gambar 2.13	Antarmuka Node-RED Pada Peramban Edge.....	13
Gambar 2.14	Modul Level Shifter.....	13
Gambar 2.15	Rangkaian Level Shifter.....	14
Gambar 3.1	Diagram Alur Tahapan Penelitian.....	15
Gambar 3.2	Perancangan Sistem.....	16
Gambar 3.3	Jaringan Sistem.....	17
Gambar 3.4	Diagram Alur Data.....	17
Gambar 3.5	Rancangan Skematik Bagian Pengirim.....	18
Gambar 3.6	Rancangan Skematik Bagian Penerima.....	20
Gambar 3.7	Diagram Alir Kerja Sistem Pengirim.....	21
Gambar 3.8	Diagram Alir Pembungkusan dan Pengiriman UDP.....	22
Gambar 3.9	Diagram Alir Kerja Sistem Penerima.....	23
Gambar 4.1	Rangkaian Pengujian Sistem Membaca Data MIDI.....	27
Gambar 4.2	Tangkapan Layar Aplikasi Node-RED.....	28
Gambar 4.3	Pengujian Sistem Membaca Data MIDI.....	28
Gambar 4.4	Tangkapan Layar Keluaran MIDI pada Pocket MIDI.....	28
Gambar 4.5	Tangkapan Layar Serial Pengujian Membaca Data MIDI.....	29
Gambar 4.6	Rangkaian Pengujian Komunikasi UDP.....	30
Gambar 4.7	Pengujian Komunikasi UDP.....	31
Gambar 4.8	Tangkapan Layar Serial Pengujian Komunikasi UDP.....	31
Gambar 4.9	Rangkaian Pengujian Waktu Tunda Satu Penerima.....	32
Gambar 4.10	Rangkaian Pengujian Waktu Tunda Dua Penerima.....	33
Gambar 4.11	Perangkat Pengujian Waktu Tunda Satu Penerima.....	33
Gambar 4.12	Perangkat Pengujian Waktu Tunda Dua Penerima.....	34

DAFTAR TABEL

Tabel 3.1	Cacah Data Byte Terhadap Status byte	23
Tabel 4.1	Pengujian Sistem Membaca Data MIDI	29
Tabel 4.2	Pengujian Waktu Tunda Sistem dengan Satu Penerima	34
Tabel 4.2	Pengujian Waktu Tunda Sistem dengan Dua Penerima	35



DAFTAR LAMPIRAN

Lampiran 1	Kode Sumber Perangkat Pengirim “midi_type.h”.....	42
Lampiran 2	Kode Sumber Perangkat Pengirim “main.cpp”	43
Lampiran 3	Kode Sumber Perangkat Penerima “main.cpp”.....	47
Lampiran 4	Kode Sumber Pengujian 1 “main.cpp”.....	49
Lampiran 5	Kode Sumber Pengujian 2 Pengirim "main.cpp"	53
Lampiran 6	Kode Sumber Pengujian 2 Penerima “main.cpp”.....	55
Lampiran 7	Kode Sumber Arduino Mega 2560 “DMS_MEGA_2560.ino”.....	57
Lampiran 8	Pengujian <i>drop</i> Paket Komunikasi UDP ESP	58



BAB I PENDAHULUAN

Bab ini berisikan latar belakang dari penelitian, rumusan masalah yang harus diselesaikan, batasan masalah yang ada pada penelitian, tujuan dari penelitian manfaat dari penelitian yang dilakukan dan sistematika penulisan dari penelitian.

1.1 Latar Belakang

MIDI adalah standar *de facto* komunikasi antara peralatan musik elektronik. Standar komunikasi yang digunakan pada industri musik ini menjawab permasalahan yang berhubungan dengan penggunaan *synthesizer*, modulator suara atau *sample*. Untuk menghubungkan satu perangkat alat musik menggunakan komunikasi MIDI, diperlukan konektor MIDI atau disebut juga konektor DIN 5 pin yang terbagi menjadi 3 jenis komunikasi, MIDI IN, MIDI OUT, dan MIDI THRU. Konektor MIDI lainnya adalah USB, FireWire (IEEE 1394), Thunderbolt, dan Ethernet (Lehrman & Tully, 2017). Sedangkan komunikasi MIDI tanpa kabel dapat menggunakan protokol Bluetooth dan Wi-Fi.

Protokol MIDI memungkinkan satu perangkat musik dapat mengirim data ke beberapa perangkat lainnya secara *daisy chain*. Koneksi *daisy chain* MIDI adalah dengan menghubungkan perangkat musik secara beruntun yaitu menghubungkan keluaran MIDI OUT sumber data ke MIDI IN perangkat pertama, keluaran MIDI THRU perangkat pertama dihubungkan ke perangkat kedua dan seterusnya. Secara teori jumlah perangkat yang dapat dihubungkan secara *daisy chain* adalah tidak terhingga, tetapi secara praktik terdapat batasan (Lehrman & Tully, 2017). Konektor MIDI USB dan Bluetooth memiliki batasan karena hanya dapat menghubungkan dua perangkat saja karena protokol yang tidak mendukung hubungan *daisy chain*. FireWire yang mendukung protokol *daisy chain* dinyatakan tidak didukung lagi karena kalah dengan protokol lainnya pada tahun 2015 (Murray, 2015). Thunderbolt yang merupakan pengganti FireWire kurang diminati produsen alat musik sebagai konektor MIDI dan lebih populer sebagai Display Port walaupun mendukung *daisy chain*. Konektor Ethernet jarang digunakan sebagai jalur komunikasi MIDI karena harus memiliki pusat pemroses data/peladen tersendiri. Konektor standar MIDI DIN memiliki batasan dalam menghubungkan perangkat secara *daisy chain*. Batasan konektor MIDI DIN adalah tidak dapat dicabangkan/diparalelkan karena karakteristik sinyal dapat berubah terutama nilai tegangan logika *high* (The MIDI Manufacturers Association, 2014).

Fungsi *daisy chain* MIDI berguna dalam membagi data MIDI ke beberapa peralatan musik, terutama *synthesizer* dan virtual *sample*. Ketika mengaransemen musik, beberapa komposer yang berada pada ruangan yang berbeda dapat menghubungkan lebih dari satu *synthesizer* untuk menghasilkan sebuah bunyi. MIDI *Sequencer* dapat terhubung dengan banyak MIDI *Sound Module* untuk menghasilkan berbagai macam suara, sehingga fungsi MIDI THRU sangat diperlukan. Studio musik memiliki tata ruang yang tersebar. Antara pemusik/komposer satu dengan lainnya berada di ruangan bahkan lantai yang berbeda. Komunikasi MIDI antara alat musik menggunakan MIDI DIN dibatasi panjang dan jumlah kabel. Semakin banyak perangkat yang terhubung maka semakin banyak dan panjang kabel

yang digunakan, sehingga dibutuhkan komunikasi berbasis tanpa kabel. Salah satu komunikasi tanpa kabel MIDI adalah melalui Wi-Fi sedangkan protokol komunikasi yang mendukung pengiriman data tersebar adalah UDP.

UDP (*User Datagram Protocol*) adalah protokol komunikasi internet yang berada di bagian lapisan *transport* TCP/IP. UDP merupakan *connectionless protocol* yang berarti tidak memerlukan *handshaking* antara kedua perangkat sebelum berkomunikasi (Kurose & Ross, 2013). Karena tidak membutuhkan *handshaking* maka UDP dapat mengirimkan pesan ke alamat IP (*Internet Protocol*) yang diinginkan termasuk alamat *broadcast* jaringan. Ukuran data UDP relatif kecil dibandingkan dengan TCP (*Transmission Control Protocol*) karena UDP memiliki *header* data yang kecil. UDP biasa digunakan sebagai media transmisi data *streaming*.

Karakteristik UDP dapat diterapkan dalam komunikasi MIDI. Data MIDI memiliki ukuran yang kecil, membutuhkan respons cepat dan dapat dikirim ke beberapa perangkat, hal itu sesuai dengan karakteristik protokol UDP yang tidak membutuhkan *handshaking* dan besar *header* data yang kecil sehingga waktu pengiriman data diharapkan relatif kecil dibandingkan dengan menggunakan TCP serta dapat dikirim ke alamat *broadcast* jaringan. Dengan menggunakan UDP *Broadcast* maka pesan MIDI yang dikirimkan dapat terkirim ke semua perangkat pada satu jaringan.

ESP32 adalah mikrokontroler yang populer di kalangan antusias dan *makker* elektronik. Mikrokontroler ini memiliki 3 UART dan blog Wi-Fi yang mendukung TCP/ IP *Stack*. Keunggulan protokol UDP dan fitur ESP32 mendorong penulis untuk mengatasi permasalahan *daisy chain* komunikasi MIDI. Diharapkan penelitian ini dapat menjawab permasalahan pengiriman jumlah perangkat yang terhubung pada MIDI THRU.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka didapatkan rumusan masalah sebagai berikut :

1. Bagaimana merancang komunikasi MIDI menggunakan ESP32?
2. Bagaimana merancang dan mengimplementasikan informasi MIDI ke dalam komunikasi protokol UDP *Broadcast*?
3. Berapa jeda yang muncul antara pengirim dan penerima?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah dibuat, maka diberikan beberapa batasan masalah sehingga pembahasan dari topik yang diangkat penulis tidak melebar. Berapa batasan masalah adalah sebagai berikut :

1. Data MIDI yang diterima berupa 3 byte/kurang pesan MIDI yang dibaca melalui komunikasi serial.
2. Salah satu ESP32 berperan sebagai pengirim.
3. Modul ESP32 lainnya berfungsi sebagai penerima.
4. Komunikasi yang terjalin antara modul ESP32 menggunakan jaringan *wireless*.
5. Jaringan Wi-Fi yang terhubung diantara modul ESP32 terisolasi dari jaringan lainnya.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Dapat merancang komunikasi MIDI menggunakan ESP32.
2. Mengimplementasikan informasi MIDI ke dalam komunikasi protokol UDP.
3. Menghitung jeda yang ditimbulkan antara pengirim dan penerima.

1.5 Manfaat Penelitian

Penelitian ini dapat bermanfaat bagi perkembangan dunia komunikasi data MIDI khususnya menggunakan teknologi UDP tanpa kabel, sehingga terjalin komunikasi yang lebih fleksibel dan mudah digunakan.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan penelitian ini sebagai berikut :

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan, tujuan, manfaat dan sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas kajian pustaka dan dasar teori yang digunakan sebagai landasan teori dalam penelitian yang dilakukan.

BAB III Metode Penelitian

Menjelaskan tentang metode yang digunakan dalam penelitian dalam tahapan penyelesaian skripsi yang meliputi studi literatur, perancangan dan pembuatan sistem, pengumpulan data, analisis, serta pengambilan kesimpulan.

BAB IV Hasil dan Pembahasan

Membahas tentang langkah dan analisis pengujian sistem serta hasil pengujian pada penelitian sistem komunikasi MIDI ini.

BAB V Kesimpulan dan Saran

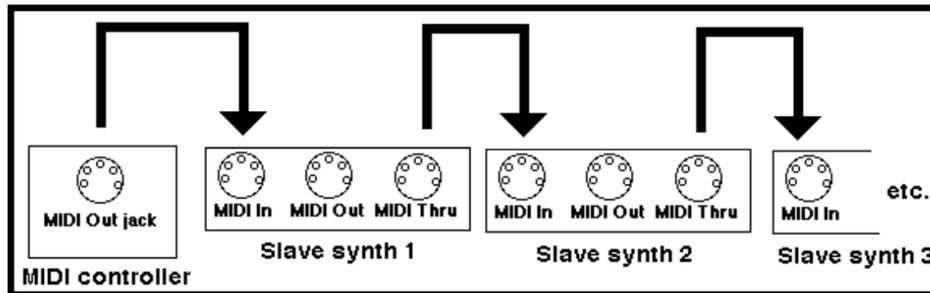
Memuat kesimpulan berdasarkan hasil yang telah didapatkan dan saran sehingga penelitian dapat dikembangkan lebih lanjut.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan teori-teori pendukung penelitian yang akan dilakukan, seperti MIDI, perangkat lunak MIDI, komunikasi UDP, ESP32 *Dev Board*, sensor loadcWi-Fi, Visual Studio Code, Node-RED, level *shifter* dan lain-lain.

2.1 MIDI

MIDI merupakan akronim dari “*Musical Instrument Digital Interface*” atau dalam bahasa Indonesia berarti antarmuka perangkat musik digital. MIDI diciptakan untuk menstandarkan komunikasi alat musik digital dan dibuat oleh The MIDI Manufacturers Association. Komunikasi MIDI tidak hanya digunakan sebagai jalur komunikasi alat musik saja, tetapi digunakan sebagai komunikasi *audio mixing*, *editing*, dan produksi serta area lain seperti gim elektronik, robot, lampu panggung, dan nada dering. Data yang dikirim melalui MIDI bukanlah data suara, melainkan deskripsi informasi musik tersebut. Penjelasan tersebut bermakna bahwa data yang dikirim adalah kode bunyi/kejadian tersebut terjadi, misalnya informasi berupa *note* nada, kecepatan, dan tekanan yang diberikan sehingga *synthesizer* dapat mereproduksi ulang.



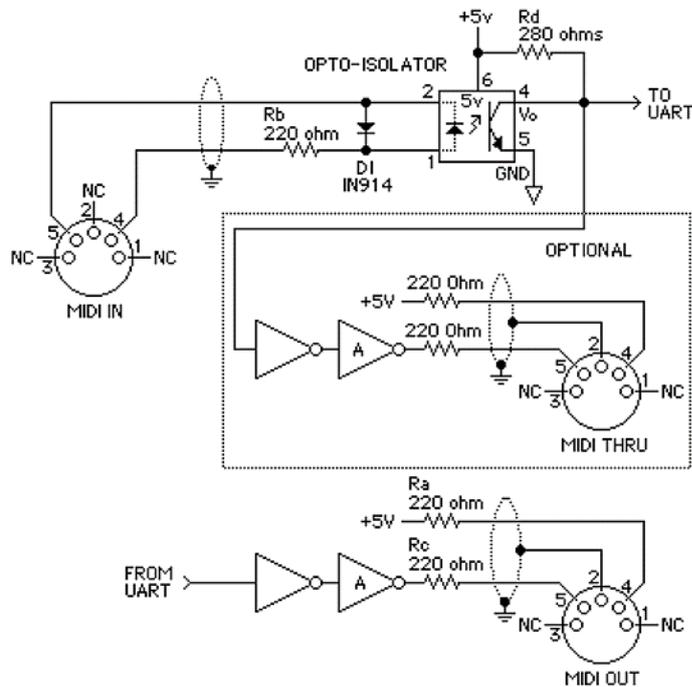
Gambar 2.1 *Daisy Chain MIDI*

Saat ini komunikasi MIDI dapat dilakukan dengan menggunakan beberapa konektor seperti MIDI DIN, USB, FireWire, Thunderbolt, Ethernet. Meskipun termasuk konektor lama, konektor MIDI DIN masih banyak dijumpai sebagai jalur komunikasi MIDI. Komunikasi MIDI terbagi menjadi 3 jenis yaitu MIDI IN, MIDI OUT, dan MIDI THRU. MIDI IN adalah jalur masuk MIDI ke perangkat, MIDI OUT adalah jalur keluar MIDI ke perangkat, sedangkan MIDI THRU adalah salinan dari jalur MIDI IN perangkat sehingga dapat dihubungkan dengan perangkat lain secara *daisy chain*. Pada konektor MIDI DIN, MIDI OUT perangkat dihubungkan dengan MIDI IN perangkat lain. **Gambar 2.1** adalah cara menghubungkan MIDI secara *daisy chain*.

Spesifikasi perangkat keras MIDI DIN adalah sebagai berikut :

1. Arus minimal komunikasi adalah 5mA.
2. Tegangan logika 5V.
3. Perangkat MIDI satu dengan lainnya terisolasi *ground*-nya.





Gambar 2.2 Rangkaian Elektronik MIDI

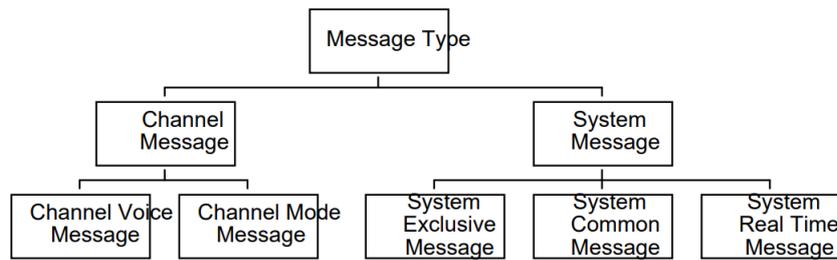
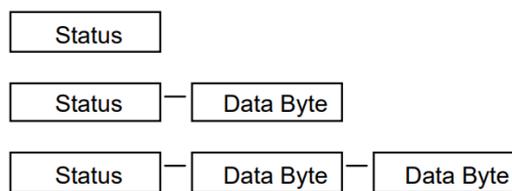
Gambar 2.2 adalah rangkaian MIDI DIN beserta keterangannya. Terdapat *opto-isolator* untuk mengisolasi rangkaian secara elektrik dan dioda untuk proteksi arus terbalik *opto-isolator*. Nilai R_a dan R_c 220 ohm berfungsi sebagai pembatas arus yang dilewatkan. Pada jalur keluaran MIDI diberikan rangkaian penyangga sehingga arus komunikasi dapat terjaga dan tidak mempengaruhi rangkaian sebelumnya.

Spesifikasi perangkat lunak MIDI :

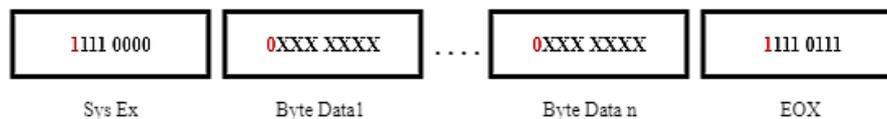
1. Bekerja pada 31,25Kbaud (+/- 1%)
2. Asinkron dengan bit mulai dan berhenti
3. 8bit per data
4. Satu pesan MIDI terdiri dari 1 hingga tidak terhingga byte data
5. Terdapat 16 kanal yang dapat diatur

2.1.1 Pesan MIDI

Pesan MIDI adalah data yang diberikan satu perangkat MIDI ke perangkat lainnya. Pesan standar MIDI memiliki panjang 1 hingga 3 byte, walaupun terdapat protokol yang memungkinkan untuk mengirim pesan dengan panjang yang tidak ditentukan. **Gambar2.3** adalah pembagian pesan MIDI dimana dibagi menjadi 2 kelompok besar yaitu pesan kanal dan pesan sistem. Pesan kanal berisi pengaturan masing-masing kanal sedangkan pesan sistem berisi pengaturan yang ditujukan ke sistem seperti waktu, pengecekan sambungan, dan komunikasi eksklusif.

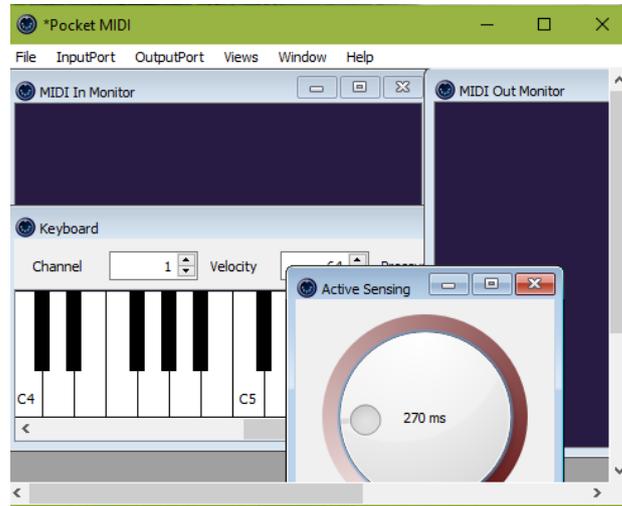
TYPES OF MIDI MESSAGES:**Gambar 2.3** Bagan Pembagian Pesan MIDI*STRUCTURE OF A SINGLE MESSAGE:**STRUCTURE OF SYSTEM EXCLUSIVE MESSAGES:***Gambar 2.4** Struktur Pesan MIDI

Gambar 2.4 adalah struktur satu pesan MIDI yang mana terdiri dari 1 hingga 3 byte data atau tidak terhitung. Karena besar byte masing-masing pesan dapat berbeda, untuk membedakan pesan MIDI satu dengan lainnya maka, setiap pesan MIDI diberikan *flag* bit. **Gambar 2.5** menunjukkan lokasi *flag* bit pada pesan MIDI. Setiap pesan MIDI diawali dengan bit 1 pada bagian byte status. Sedangkan byte lainnya akan diawali dengan bit 0.

Pesan MIDI**Pesan MIDI Sistem Eksklusif****Gambar 2.5** Byte Pesan MIDI**2.1.2 Pocket MIDI**

Perangkat lunak Pocket MIDI adalah perangkat yang berfungsi untuk memantau, memetakan dan memfilter data MIDI. Terdapat fitur untuk membuat data MIDI sesuai

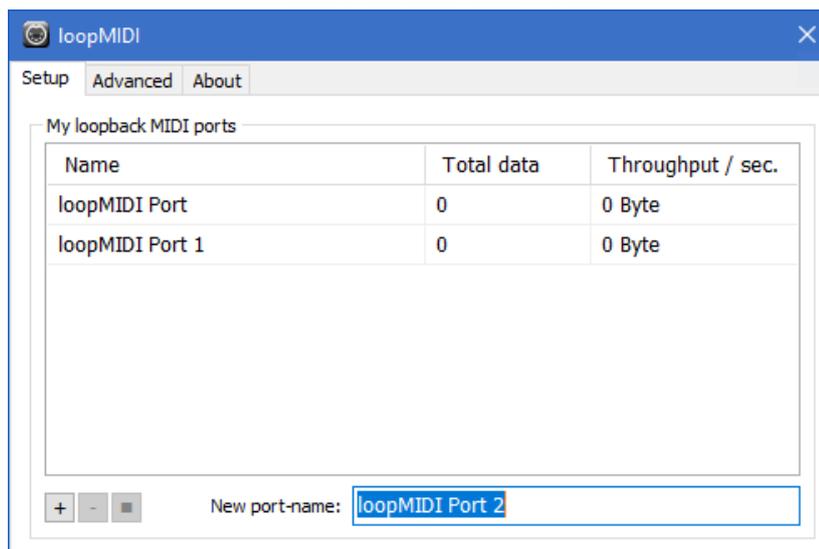
dengan kehendak melalui papan ketik atau fungsi yang tersedia. Secara resmi, perangkat ini dapat berjalan pada sistem operasi Windows XP hingga Windows 10 dan bekerja pada 32bit/64 bit. Program ini gratis untuk penggunaan perorangan dan bukan komersial dan dapat diunduh melalui laman <https://morson.jp/pocketmidi-webpage>.



Gambar 2.6 Antarmuka MIDI-OX

2.1.3 LoopMIDI

LoopMIDI adalah program yang menyimulasikan (virtual) *port* MIDI yang berjalan pada sistem operasi Windows XP hingga Windows 10 32bit/64bit. Fungsi perangkat lunak ini adalah untuk menjembatani komunikasi MIDI antara perangkat lunak satu dengan lainnya dengan membuat *port loopback* MIDI. Perangkat lunak yang dibuat oleh Tobias Erichsen ini bekerja dengan menggunakan *driver virtualMIDI* yang juga dibuat oleh dia. Program ini gratis untuk penggunaan perorangan dan bukan komersial. Program ini dapat diunduh melalui laman <http://www.tobias-erichsen.de/software/>.



Gambar 2.7 Antarmuka loopMIDI

2.1.4 USB-MIDI

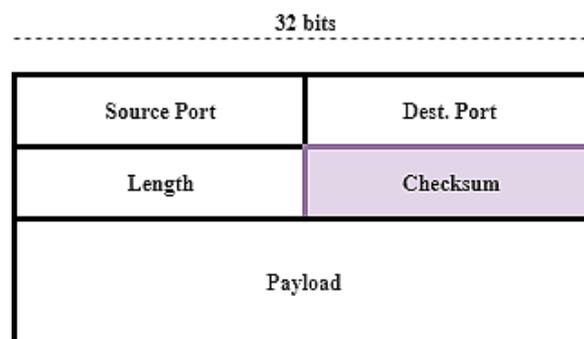
USB-MIDI adalah perangkat keras yang berfungsi sebagai MIDI IN dan MIDI OUT perangkat PC melalui sambungan USB ke konektor MIDI DIN. Konektor USB dihubungkan ke PC dan konektor MIDI DIN dihubungkan ke perangkat MIDI. *Driver* yang disediakan berjalan pada sistem operasi GNU/Linux, Mac OS, dan Windows.



Gambar 2.8 USB-MIDI

2.2 UDP

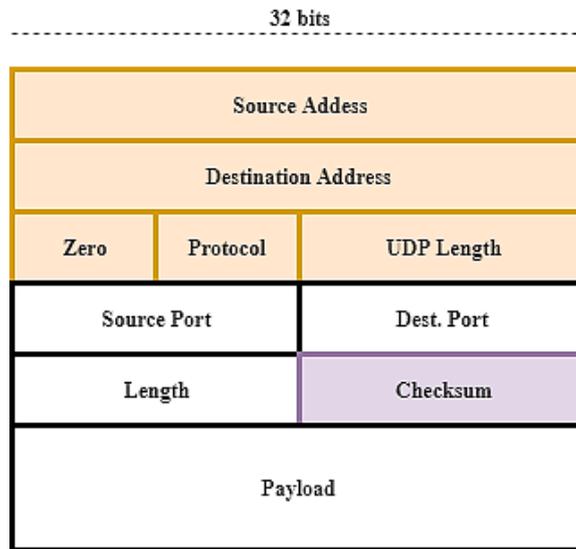
User Datagram Protokol atau UDP (RFC 768) adalah salah satu dari protokol internet yang berada pada lapisan *transport*. Untuk berkomunikasi melalui internet menggunakan UDP, lapisan aplikasi komputer dapat mengirim pesan yang berbentuk seperti datagram. UDP adalah komunikasi yang tidak membutuhkan koneksi yang bermakna bahwa UDP tidak perlu menghubungkan tujuan apabila ingin melakukan komunikasi. Pesan UDP akan langsung dikirimkan ke alamat tujuan tanpa memastikan apakah alamat tujuan ada/siap untuk menerima pesan.



Gambar 2.9 Header UDP

UDP memiliki ukuran yang kecil dibandingkan TCP (*Transmission Control Protocol*) ukuran *header* UDP adalah 8 byte yang terdiri dari 4 kolom yaitu sumber *port*, tujuan *port*, panjang pesan, dan *checksum*. **Gambar 2.9** menggambarkan struktur pesan UDP. Sumber (*Source*) *port* berisi 8 bit *port* sumber yang berfungsi sebagai pemberi tahu *port* ke alamat tujuan apabila ingin menjawab pesan UDP yang dikirim. Sumber *port* bernilai 0 apabila tidak ada sumber *port* yang ingin disampaikan. Apabila sumber *port* adalah peladen maka nilainya adalah *port* yang sudah diketahui, apabila klien maka nilai *port* tergantung aplikasi yang digunakan. Tujuan (*Destination*) *port* berisi 8 bit *port* tujuan dan bersifat wajib diisi. Panjang pesan berisi panjang pesan termasuk *header* dan pesan. Panjang pesan termasuk *header* minimal 8 bit dan maksimal 65.507 byte pada IPv4 (8 byte *header*, 20 byte IP

header). *Checksum* digunakan untuk menentukan nilai rusak atau tidaknya data *header*. *Checksum* bersifat opsional dan bernilai 0 apabila tidak digunakan.



Gambar 2.10 Header UDP IPv4

Pada IPv4 (*Internet Protokol v4*), *header* UDP memiliki tambahan sebesar 20 byte *header*. **Gambar 2.10** menunjukkan pembagian kolom *header* UDP IPv4. Kolom pertama adalah alamat sumber IPv4 sebesar 4 byte, dilanjutkan dengan alamat tujuan IPv4 sebesar 4 byte. Kolom selanjutnya berupa 1 byte bernilai kosong dan dilanjutkan 1 byte protokol yang disediakan IPv4, serta dilanjutkan dengan panjang pesan UDP sebesar 2 byte.

2.3 ESP32 Dev Board

ESP32 adalah sebuah komponen *chip* terintegrasi yang didesain dan diproduksi oleh Espressif. *Chip* ini dibuat untuk menjawab permasalahan komunikasi mikrokontroler yang berhubungan dengan Wi-Fi. Bahasa pemrograman yang didukung mikrokontroler ini adalah C, MicroPython, JavaScript dan LUA. Framework pemrograman menggunakan ESP IDF dan Arduino sehingga dapat menggunakan Arduino IDE atau yang setara.

Modul Wi-Fi ini merupakan SoC (*System on Chip*) yang terintegrasi dengan protokol *stack* TCP/IP, sehingga memungkinkan mikrokontroler untuk mengakses jaringan Wi-Fi. Salah satu papan pengembang yang menggunakan IC (*Integrated Circuit*) ESP32 adalah ESP32 *Dev Board*. Papan ini sudah dilengkapi dengan label pin, USB-Serial, regulator tegangan dan LED indikator.

ESP32 *Dev Board* memiliki GPIO (*General Purpose Input Output*) sebanyak 30 pin. Untuk memprogram modul ini dapat melalui koneksi USB yang sudah tersedia. Seperti terlihat pada **Gambar 2.11**, modul ESP32 *Dev Board* sudah dilengkapi dengan antena dengan IC dilindungi sehingga terlindung dari interferensi frekuensi radio lainnya.



Gambar 2.11 Papan ESP32 Dev Board

Spesifikasi dari ESP32 *Dev Board* adalah sebagai berikut :

1. Besar SRAM 512 kB.
2. SPI *flash* 4MB.
3. Tegangan kerja masukan 3.3 VDC.
4. Memiliki LED indikator.
5. Jaringan Wi-Fi pada 802.11 b/g/n hingga 150kbps.
6. Bluetooth 4.2.
7. Menggunakan sistem Wi-Fi *Direct* (P2P), *soft-AP*.
8. Memiliki pin 3 UART.
9. 10-bit ADC.
10. 8-bit DAC.
11. *Dual core* Xtensa 32bit dengan frekuensi CPU hingga 240MHz.
12. *Interface* : SPI, I²C, I²S.
13. SPI FLASH 4MB.
14. GPIO I_{OH} 40mA I_{OL} 26 mA.
15. GPIO R *Pull-Up* R *Pull-Down* 46 kOhm.

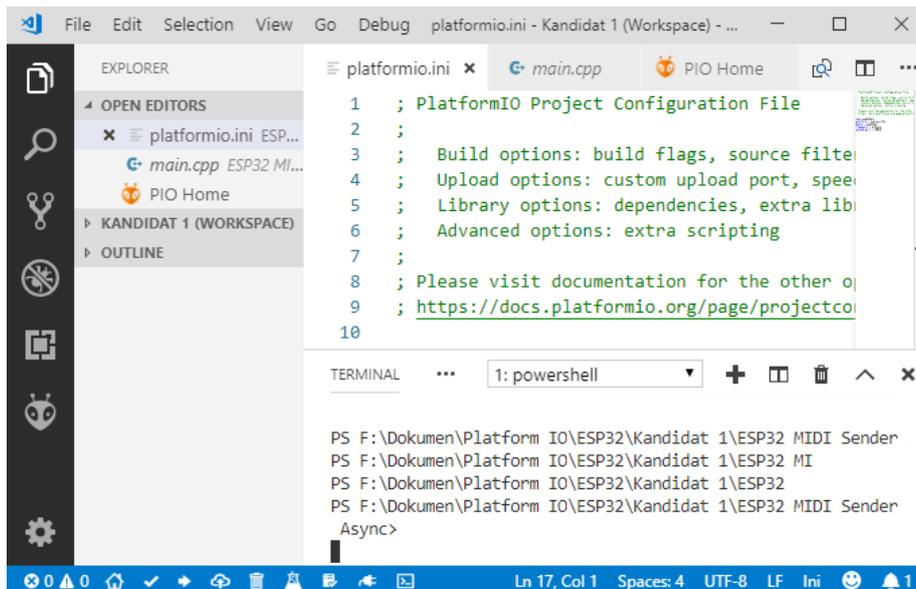
2.4 Wi-Fi

Wireless Local Access Network atau WLAN adalah sinonim dari Wi-Fi. Wi-Fi sebenarnya merupakan sebuah merek dagang dari Wi-Fi Alliance yang bergerak dibidang standarisasi jaringan WLAN. Wi-Fi mengacu pada standar IEEE 802.11. Standar ini adalah protokol cara berkomunikasi antara perangkat tanpa kabel pada satu jaringan. Komunikasi tanpa kabel rentan terhadap kebocoran data, sehingga Wi-Fi Alliance membuat sebuah standar pengamanan berupa WEP (*Wired Equivalent Privacy*), WPA (*Wi-Fi Protected Access*), WPA2, WPS (*Wi-Fi Protected Setup*), dan WPA3 yang sedang dikembangkan.

Terdapat dua jenis komunikasi Wi-Fi yaitu, Ad-Hoc dan Infrastruktur. Komunikasi Ad-Hoc adalah jenis komunikasi *peer-to-peer* antara perangkat. Perangkat satu dapat terhubung dengan perangkat lain secara langsung. Komunikasi Infrastruktur atau sering disebut *Access Point* adalah komunikasi yang membutuhkan pusat pengolah lalu lintas data. Jumlah perangkat yang terhubung ke *Access Point* disebut juga *Client* lebih banyak dari pada jenis *Ad-Hoc*. Untuk dapat terhubung ke *Access Point* dibutuhkan SSID (*Service set Identifier*).

2.5 Visual Studio Code

Visual Studio Code adalah perangkat lunak yang terintegrasi yang berfungsi sebagai tempat pengembangan dan membangun perangkat lunak atau *Integrated Development Environment* (IDE). Perangkat lunak yang terbuka ini ditujukan untuk membuat program perangkat yang dikembangkan untuk berbagai keperluan. Program ini dapat berjalan pada Windows, Mac OS X, dan GNU/Linux. Terdapat fitur ekstensi untuk menambahkan berbagai macam IDE atau *Framework* lain. Program dapat diunduh melalui laman <https://code.visualstudio.com>.



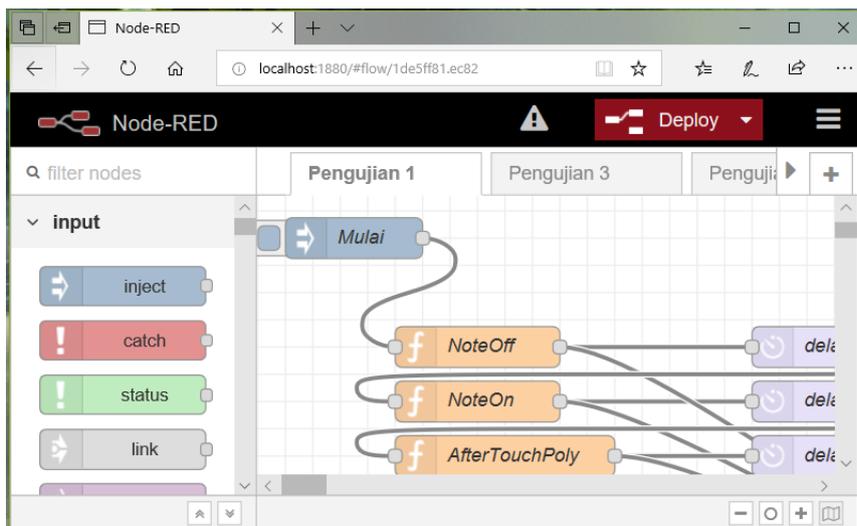
Gambar 2.12 Antarmuka Visual Studio Code

2.5.1 Ekstensi Platform IO

Platform IO atau biasa disebut PIO adalah IDE yang dikembangkan untuk kebutuhan *Internet of Things* (IoT). PIO dapat bekerja pada sistem operasi GNU/Linux atau sebagai ekstensi Visual Studio Code. IDE ini mendukung pemrograman bahasa C++ dengan *framework* yang beragam. Proses penambahan PIO pada Visual Studio Code adalah dengan masuk pada tab *Extensions* dan mencari Platform IO pada kolom pencarian.

2.6 Node-RED

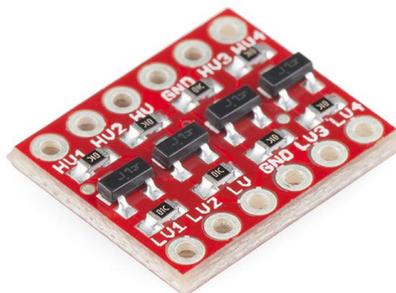
Node-RED adalah perangkat pemrograman untuk menghubungkan beberapa perangkat keras secara bersamaan dan mendukung komunikasi internet. Node-RED berjalan pada Node.js dengan pemrograman JavaScript berkonsep alur (*flow*). Pemrograman dilakukan melalui laman *browser* dan bekerja layaknya peladen. Proses instalasi Node-RED dapat dilakukan dengan mengunduh dan menjalankan berkas pada laman <http://nodered.org>.



Gambar 2.13 Antarmuka Node-RED pada Peramban Edge

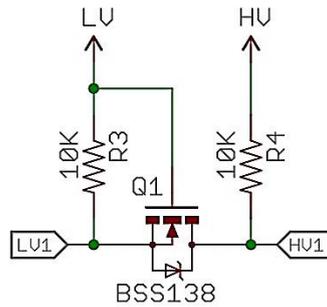
2.7 Level Shifter

Modul pengubah tegangan logika tegangan atau disebut level *shifter* dibuat oleh SparkFun. Modul ini dapat bekerja secara dua arah dan sering dijumpai pada penurun level tegangan I2C atau Serial. **Gambar 2.14** adalah modul level *shifter* 4 kanal. LV adalah tegangan logika *high* yang rendah sedangkan HV adalah tegangan logika *high* yang tinggi.



Gambar 2.14 Modul Level Shifter

Skematik rangkaian level *shifter* dapat dilihat pada **Gambar 2.15**. Mosfet yang digunakan adalah kanal N BSS138 dengan tambahan resistor *pull-up* pada *drain* dan *source* sebesar 10kohm. LV1 adalah jalur logika bertegangan rendah, LV adalah sumber tegangan rendah, HV1 adalah jalur logika bertegangan tinggi, dan HV adalah sumber tegangan tinggi. R3 terhubung di antara LV dengan *source*/LV1 sedangkan R4 terhubung di antara HV dengan *drain*/HV1. *Gate* mosfet selalu terhubung dengan LV.



Gambar 2.15 Rangkaian *Level Shifter*

Prinsip kerja rangkaian ini dijelaskan dalam 3 kondisi berikut:

1. Kondisi LV1 dan HV1 tidak berlogika *low*, maka nilai LV1 akan setara dengan LV karena resistor R3. Sehingga nilai V_{GS} mosfet tidak mencapai batas *threshold* yang menyebabkan *drain* dan *source* mosfet tidak terhubung. Hal tersebut membuat *drain*/HV1 bertegangan setara HV akibat resistor R4. Jadi LV1 dan HV1 sama-sama berlogika *high* dengan level tegangan berbeda.
2. Kondisi LV1 berlogika *low*, maka *source* mosfet akan memiliki tegangan lebih rendah dari *gate* yang menyebabkan V_{GS} melewati batas *threshold*. Hal tersebut membuat *source* dan *drain* terhubung, sehingga arus dari HV1 dapat mengalir ke *source*. Jadi LV1 dan HV1 akan sama-sama dalam kondisi *low*.
3. Kondisi HV1 berlogika *low*, pertama-tama akibat dioda pada *drain-substrate* menimbulkan arus dari *source* ke *drain*. Hal tersebut berakibat tegangan *source* lebih rendah dari *gate* dan berakibat V_{GS} melewati batas *threshold* sehingga *source* dan *drain* terhubung. Muncul arus dari LV1 ke HV1. Jadi LV1 dan HV1 akan sama-sama dalam kondisi *low*.

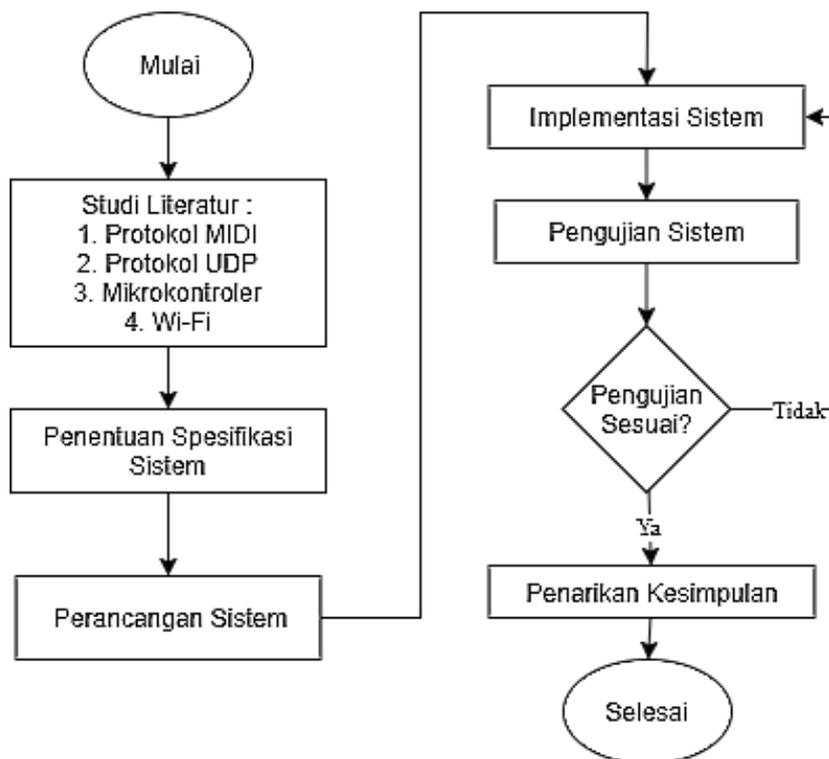
BAB III METODE PENELITIAN

Pada bab ini penulis menjelaskan langkah-langkah yang akan dilakukan dalam penyusunan skripsi meliputi metode penelitian, studi literatur, perancangan dan penentuan spesifikasi sistem, langkah kerja sistem, pengujian dan analisis sistem, dan analisa hasil dan kesimpulan.

3.1 Metode Penelitian

Penelitian ini diawali dengan studi literatur yang memiliki topik terkait dengan penelitian yang berdasar pada tinjauan pustaka dan dasar teori dari penelitian sebelumnya. Penelitian ini bersifat implementasi dari protokol UDP sebagai sarana komunikasi antara perangkat MIDI. Tahap selanjutnya adalah tahap analisa spesifikasi alat berupa spesifikasi sistem, spesifikasi perangkat lunak, dan spesifikasi perangkat keras. Tahap perancangan dilaksanakan berdasarkan desain dari blog diagram sistem yang telah dirancang dan diwujudkan dalam bentuk purwarupa yang selanjutnya dapat dilakukan pengujian.

Untuk mempermudah dalam penelitian, dibuat alur metode penelitian yang merupakan langkah-langkah yang ditempuh dalam menyelesaikan penelitian. Adapun alur metode penelitian yang dilakukan adalah sebagai berikut :



Gambar 3.1 Diagram Alur Tahapan Penelitian

3.2 Studi Literatur

Studi literatur didasarkan pada hasil pemahaman dasar teori yang terdiri dari konsep-konsep berikut ini :

1. Protokol MIDI

Studi literatur yang dilakukan terkait dengan komunikasi yang terjadi dalam MIDI, termasuk perangkat/media yang digunakan. Sumber terbaik dalam studi ini berasal dari dokumentasi resmi standar MIDI. Pokok yang dipelajari adalah tipe data komunikasi, topologi komunikasi, dan karakteristik fisik data MIDI.

2. Protokol UDP

Pada tahap ini dilakukan kajian literatur yang berkaitan dengan konsep protokol UDP, yang dapat dilakukan dengan mencari dokumentasi dasar protokol UDP. Dan mempelajari penerapan protokol UDP *Broadcast* pada sebuah jaringan.

3. ESP32

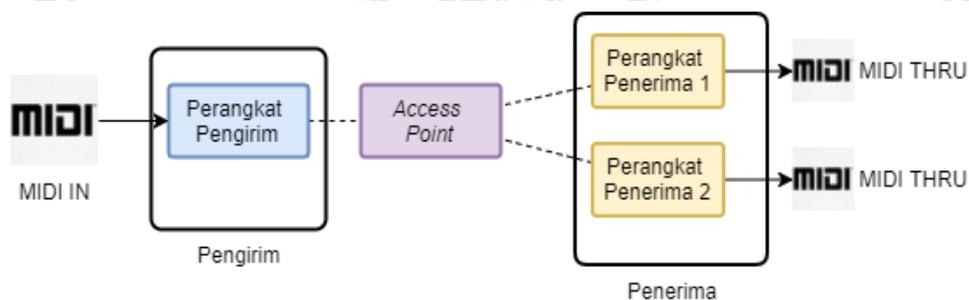
Kajian literatur yang dilakukan bertujuan untuk bahan implementasi protokol UDP, komunikasi perangkat menggunakan jaringan dan komunikasi dengan protokol MIDI.

4. Wi-Fi

Fungsi dari kajian tentang Wi-Fi adalah supaya komunikasi yang terjalin antara perangkat berupa tanpa kabel.

3.3 Perancangan dan Penentuan Spesifikasi Sistem

Berdasarkan studi literatur dan tinjauan pustaka untuk merancang komunikasi MIDI melalui UDP pada ESP32 dapat direncanakan dalam infrastruktur berikut ini:

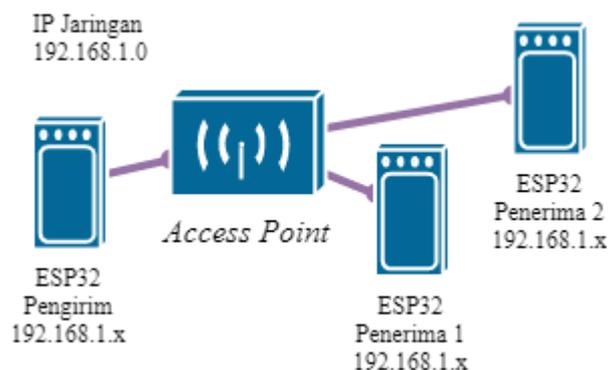


Gambar 3.2 Perancangan Sistem

Gambar 3.2 menjelaskan bagian-bagian dari sistem yang terdiri dari bagian pengirim dan penerima. Bagian pengirim berfungsi sebagai pembaca/penerjemah komunikasi MIDI ke bentuk paket UDP melalui sambungan Wi-Fi menuju perangkat penerima. Bagian penerima berfungsi sebagai penerima paket UDP dari perangkat pengirim melalui sambungan Wi-Fi dan mengubahnya kembali ke bentuk sinyal MIDI.

Perancangan sistem pada jaringan area lokal dapat dilihat pada **Gambar 3.3**. Sistem berada pada jaringan tersendiri dengan menggunakan komunikasi Wi-Fi yang disediakan *Access Point*. Protokol yang digunakan adalah TCP/IP dengan pengaturan IP secara *Dinamic*

Host Configuration Protocol (DHCP). Pada penelitian ini, jumlah perangkat penerima adalah dua yang mewakili lebih dari satu *node*.



Gambar 3.3 Jaringan Sistem

Dari perancangan sistem tersebut, bentuk aliran data yang terjadi pada penelitian ini dijelaskan pada **Gambar 3.4**. Sumber data berasal dari perangkat MIDI berupa *raw* MIDI dan diteruskan ke bagian pengubah beberapa pesan MIDI menjadi 1 paket UDP. Paket UDP disebar ke jaringan melalui jaringan Wi-Fi oleh *Access Point* ke bagian penerima. Paket UDP yang diterima oleh bagian penerima diubah kembali menjadi pesan MIDI. Pesan MIDI diteruskan ke perangkat MIDI tujuan.



Gambar 3.4 Diagram Alur Data

Untuk mempermudah proses pemodelan dan perancangan sistem maka, perancangan sistem dibagi menjadi dua bagian yaitu bagian pengirim dan penerima. Berikut ini adalah spesifikasi sistem bagian pengirim dan penerima:

3.3.1 Spesifikasi Bagian Pengirim

Bagian pengirim berhubungan langsung dengan data MIDI yang berasal dari perangkat MIDI, sehingga bagian pengirim disesuaikan dengan spesifikasi komunikasi MIDI. Berdasarkan studi literatur, komunikasi MIDI memiliki sistem yang hampir serupa dengan komunikasi serial dengan *baudrate* sebesar 31250 baud. Sedangkan karakteristik tegangan yang digunakan adalah 5V.

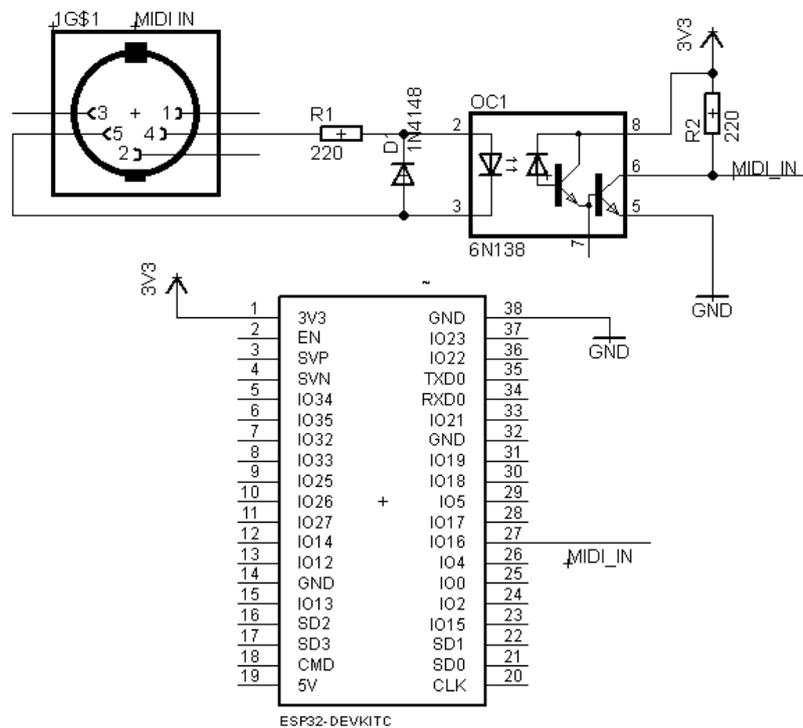
Fungsi lain dari bagian pengirim adalah dapat mengolah pesan MIDI menjadi paket UDP. Komunikasi UDP selanjutnya dikirimkan melalui sambungan Wi-Fi. Perangkat pengirim dapat mengirim data ke *Access Point*. Sehingga dibutuhkan perangkat yang mendukung komunikasi serial dan manajemen Wi-Fi serta protokol TCP/IP.

Berdasarkan spesifikasi bagian pengirim, maka dipilihlah perangkat yang dibutuhkan dalam pembuatan purwarupanya. Berikut ini adalah perangkat yang dibutuhkan sebagai pembuatan purwarupa :

1. Perangkat Keras Bagian Pengirim

Adapun perangkat keras yang akan digunakan pada perancangan sistem bagian pengirim sebagai berikut :

- Modul ESP32 Dev board yang berfungsi sebagai Wi-Fi *Client* dan pengolah data serial/MIDI ke UDP.
- Konektor MIDI DIN sebagai konektor fisik MIDI.
- Perubah level logika TTL-CMOS menggunakan *Opto-isolator* dan rangkaian pendukungnya yang sesuai dengan standar yang ditetapkan MIDI yang berfungsi sebagai rangkaian pengondisi sinyal.
- LED yang berfungsi sebagai indikator alat yang berada pada modul ESP32.



Gambar 3.5 Rancangan Skematik Bagian Pengirim

Modul ESP32 *Dev Board* adalah modul komunikasi yang berisikan papan ESP32-WROOM-32 yang sudah terintegrasi dengan komponen pendukungnya seperti IC komunikasi USB-Serial dan regulator tegangan. **Gambar 3.5** Menunjukkan skematik rangkaian pengirim yang berisi modul ESP32 *Dev Board* beserta rangkaian pengondisi sinyal. Tegangan masukan bagian pengirim berasal dari konektor USB pada ESP32 *Dev Board*. Rangkaian pengondisi sinyal disuplai dengan tegangan 3,3V yang berasal dari modul ESP32 *Dev Board*.

Komunikasi MIDI mewajibkan setiap perangkat MIDI yang terhubung terisolasi dengan perangkat lainnya, termasuk jalur *ground* . Sehingga dibutuhkan rangkaian isolator berupa *opto-isolator*. Rangkaian *opto-isolator* yang digunakan dalam penelitian ini didasarkan pada contoh rangkaian standar MIDI yang berada pada buku spesifikasi MIDI. Yang membedakan adalah *opto-isolator* disuplai dengan tegangan 3,3V sehingga keluaran

rangkaian *opto-isolator* adalah logika 3,3V. Dioda D1 berfungsi sebagai proteksi *opto-isolator* terhadap arus terbalik, resistor R1 220 ohm sebelum dioda berfungsi sebagai pembatas arus *opto-isolator*, sedangkan resistor R2 220 ohm berfungsi sebagai resistor *pull-up* keluaran rangkaian pengondisi sinyal. Keluaran rangkaian pengondisi sinyal terhubung dengan pin *General Purpose Input Output* (GPIO) 16 yang merupakan jalur masukan serial 2 modul ESP32 *Dev Board*.

2. Perangkat Lunak Bagian Pengirim

Adapun perangkat lunak yang akan digunakan pada perancangan sistem bagian pengirim sebagai berikut :

- a. Visual Studio Code dengan tambahan ekstensi Platform IO berfungsi sebagai tempat memprogram dan mengompilasi modul ESP32.
- b. *Framework* yang digunakan adalah Arduino ESP32 RTOS.
- c. Pustaka Wi-Fi ESP32 Arduino yang berisi pustaka pendukung komunikasi Wi-Fi pada modul ESP32.
- d. Pustaka UDP *Async* Arduino yang berisi pustaka komunikasi UDP pada ESP32.

3.3.2 Spesifikasi Bagian Penerima

Bagian penerima berhubungan dengan perangkat MIDI selanjutnya, sehingga dibutuhkan keluaran yang sesuai dengan spesifikasi MIDI yaitu *baudrate* sebesar 31250 baud dengan tegangan logika 5V. Sebagian besar mikrokontroler memiliki logika 3,3V sedangkan keluaran MIDI berlogika 5V maka dibutuhkan rangkaian pengubah logika dari 3,3V menjadi 5V.

Bagian penerima berkomunikasi dengan bagian pengirim menggunakan komunikasi Wi-Fi sehingga, dibutuhkan spesifikasi perangkat yang dapat berkomunikasi dengan Wi-Fi dan mendukung protokol TCP/IP. Paket UDP yang diterima bagian penerima dikembalikan lagi ke bentuk pesan MIDI sehingga spesifikasi perangkat juga harus mendukung komunikasi UDP dan serial.

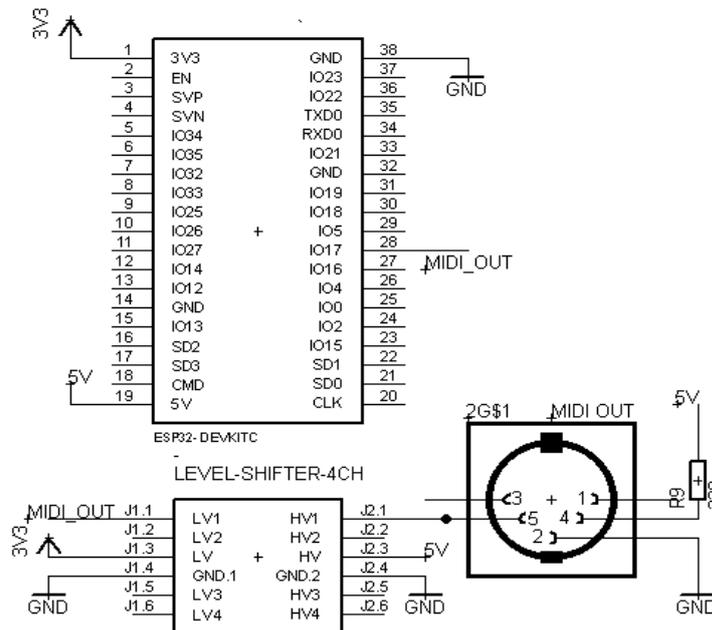
Berdasarkan spesifikasi bagian penerima, maka perangkat yang dipilih pada penelitian ini adalah sebagai berikut :

1. Perangkat Keras Bagian Penerima

Perangkat keras yang dibutuhkan adalah sebagai berikut :

- a. Modul ESP32 *Dev Board* yang berfungsi sebagai Wi-Fi *Client* dan pengolah UDP ke MIDI/serial.
- b. Konektor MIDI DIN sebagai konektor fisik MIDI.
- c. Perubah level logika CMOS/TTL berupa modul *logic converter* yang menggunakan MOSFET kanal N BSS138 yang berfungsi sebagai rangkaian pengondisi sinyal.

- d. LED yang berfungsi sebagai indikator alat yang berada pada modul ESP32.



Gambar 3.6 Rancangan Skematik Bagian Penerima

Pada **Gambar 3.6** diperlihatkan rangkaian skematik modul ESP32 *Dev Board*. Sama dengan bagian pengirim, modul ESP32 *Dev Board* sudah memiliki rangkaian regulator tegangan sehingga dapat menyuplai modul *Level Shifter* dengan tegangan 5V dan 3,3V. Keluaran ESP32 *Dev Board* adalah pin GPIO 17 yang merupakan jalur keluaran serial 2. Rangkaian *Level Shifter* penerima menggunakan modul yang sudah jadi yang didasarkan pada MOSFET kanal N BSS138 dan resistor 10k ohm. Keluaran dari rangkaian modul tersebut memiliki karakteristik tegangan logika 5V. Diagram konektor MIDI disesuaikan dengan standar pin MIDI DIN.

2. Perangkat Lunak Bagian Penerima

Adapun perangkat lunak yang akan digunakan pada perancangan sistem bagian pengirim sebagai berikut :

- Visual Studio Code dengan tambahan ekstensi Platform IO berfungsi sebagai tempat memprogram dan mengompilasi modul ESP32.
- Pustaka Wi-Fi ESP32 Arduino yang berisi pustaka pendukung komunikasi Wi-Fi pada modul ESP32.
- Pustaka UDP Async Arduino yang berisi pustaka komunikasi UDP pada ESP32.

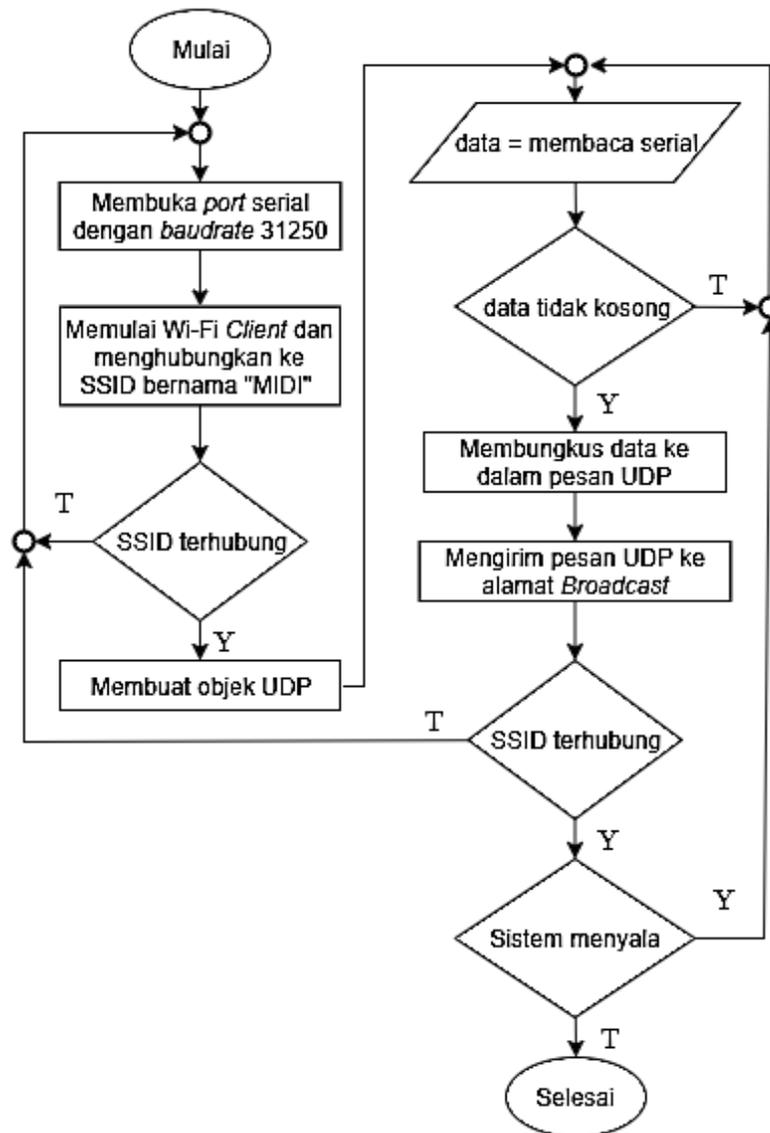
3.3.3 Pengaturan *Access Point*

Wi-Fi *Access Point* berfungsi untuk menjembatani pengirim dan penerima memiliki spesifikasi minimal mendukung protokol Wi-Fi 802.11 b/g/n pada 2,4GHz. Pengaturan

router diaktifkan dengan IP jaringan lokal 192.168.4.0. SSID *Access Point* adalah “MIDI” dengan kata sandi “MIDIMIDI” dan menggunakan pengaturan kanal otomatis.

3.4 Langkah Kerja Sistem

Secara garis besar, langkah-langkah kerja sistem bagian pengirim dijelaskan dalam diagram alir berikut :



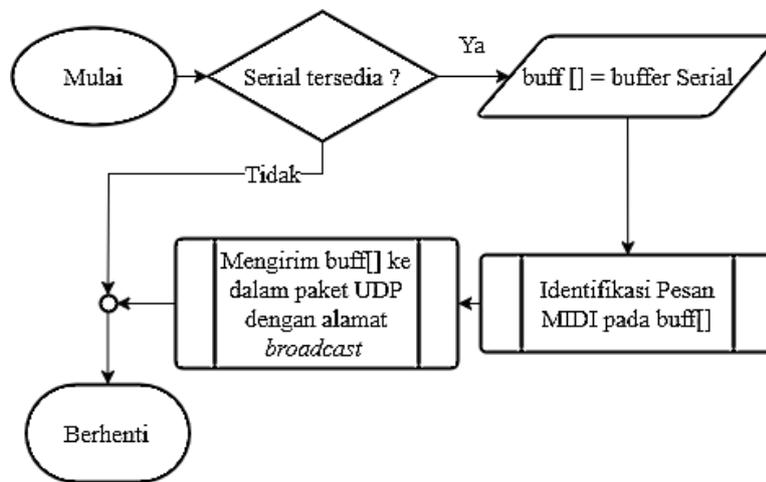
Gambar 3.7 Diagram Alir Kerja Sistem Pengirim

Diagram alur sistem pada bagian perangkat pengirim dijelaskan lebih lanjut melalui poin-poin berikut :

1. Sistem memulai dengan membuka *port* serial dengan *baudrate* sebesar 31250 bps.
2. Sistem memulai proses menghubungkan Wi-Fi ke jaringan dengan SSID bernama “MIDI”.
3. Sistem memastikan bahwa sistem sudah terkoneksi dengan SSID yang sudah ditentukan. Apabila tidak, maka akan *restart*.
4. Sistem membuat objek UDP.

5. Sistem membaca nilai *buffer* serial dan memasukkan ke variabel bernama “data”.
6. Apabila “data” bernilai kosong maka proses pembacaan serial diulang kembali
7. Apabila “data” bernilai tidak kosong maka, “data” dibungkus ke dalam pesan UDP.
8. Pesan UDP dikirim ke alamat *broadcast*.
9. Sistem akan memastikan bahwa sistem masih terhubung ke SSID yang sudah ditentukan.
10. Apabila sistem masih menyala maka sistem akan mengulang kembali ke tahap pembacaan serial.

Pada protokol UDP dikenal dengan *port*/saluran yang digunakan untuk mengirim dan mendengarkan paket UDP. *Port* yang digunakan pada penelitian ini adalah 1112. Pemilihan angka *port* adalah acak karena jaringan yang digunakan tertutup.



Gambar 3.8 Diagram Alir Pembungkusan dan Pengiriman UDP

Gambar 3.8 adalah mekanisme pembungkusan dan pengiriman data serial menjadi paket UDP. Proses pembungkusan pesan MIDI menjadi paket UDP dilakukan dengan memfilter masing-masing byte serial pada *buffer* serial dan mengidentifikasi berbagai macam pesan MIDI. Pesan MIDI yang sudah teridentifikasi dimasukkan ke dalam *array* tempat penampungan dan dibungkus ke dalam paket UDP.

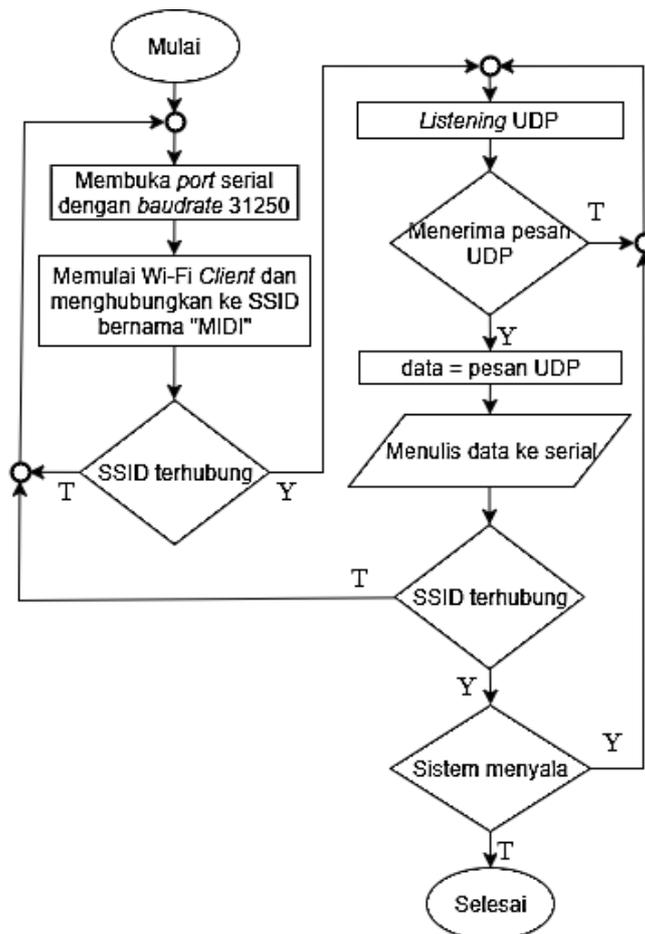
Satu pesan MIDI terdiri dari 1 *byte* status *byte* dan atau tanpa 1 atau 2 *byte* data *byte*. Juga terdapat pesan MIDI (*MIDI Exclusive*) yang terdiri dari 1 status *byte* pada awal pesan, *byte* data dengan jumlah yang tidak terhingga dan diakhiri dengan 1 status *byte* penutup. **Tabel 3.1** berisi cacah data *byte* berdasarkan status *byte*. Dalam penelitian ini implementasi identifikasi pesan MIDI dilakukan semuanya termasuk *MIDI Exclusive*, tetapi tidak dilakukan pengujian karena keterbatasan perangkat keras.

Tabel 3.1 Cacah Data Byte Terhadap Status byte

Status byte	Cacah Data byte
Start, Continue, Stop, Clock, ActiveSensing, SysReset, TuneRequest, EndOfSysExclusive(EOX)	0
ProgramChange, AfterTouchCh, TimeCodeQuarterFrame, SongSelect	1
NoteOn, NoteOff, ControlChange, PitchBend, AfterTouchPoly, SongPotition	2
SysExclusive	∞

Proses identifikasi pesan MIDI dilakukan dengan melihat bit pertama pada masing-masing byte *buffer* serial. Apabila bit pertama pada sebuah byte adalah 1 maka byte tersebut adalah status MIDI. Apabila bit pertama pada sebuah byte adalah 0 maka byte tersebut adalah data MIDI. Proses identifikasi dan pembungkusan pesan MIDI dapat dilihat pada lampiran program **halaman 43**.

Sedangkan langkah-langkah kerja sistem bagian penerima dijelaskan dalam diagram alir berikut :



Gambar 3.9 Diagram Alir Kerja Sistem Penerima

Penjelasan lebih lanjut mengenai diagram alur pada sistem penerima adalah sebagai berikut :

1. Sistem dimulai dengan membuka *port* serial dengan *baudrate* 31250 bps.
2. Sistem memulai proses menghubungkan Wi-Fi ke jaringan dengan SSID bernama “MIDI”.
3. Sistem memastikan bahwa sistem sudah terkoneksi dengan SSID yang sudah ditentukan.
4. Apabila sudah terhubung dengan jaringan yang sudah ditentukan, maka sistem memulai proses *UDP Listening*.
5. Apabila ada data UDP yang diterima, maka data UDP tersebut dimasukkan ke variabel bernama “data”.
6. Sistem menulis data serial dengan nilai dari variabel “data”.
7. Sistem akan memastikan bahwa sistem masih terhubung ke SSID yang sudah ditentukan.
8. Apabila sistem masih menyala maka sistem akan mengulang kembali ke tahap *listening* UDP.

3.4.1 Instalasi dan Konfigurasi Program

Proses instalasi dan konfigurasi program yang digunakan adalah berikut:

1. Visual Studio Code
IDE yang digunakan adalah Visual Studio Code. Proses instalasi pada Windows dapat dilakukan dengan mengunduh pada laman <https://code.visualstudio.com>. Ekstensi Platform IO ditambahkan dengan mencari Platform IO pada tab *extentions*. Instalasi *source* Arduino ESP32 RTOS dengan cara membuat proyek yang menggunakan modul ESP32. Secara otomatis akan mengunduh *source* ESP32. Penentuan *port* serial disesuaikan pada *port* serial masing-masing perangkat keras dan dapat diatur pada berkas “platformio.ini” dengan variabel “upload_port” .
2. Instalasi dan Konfigurasi Node-RED
Node-RED adalah Node.js yang berfungsi untuk membuat/menyimulasi pesan MIDI pada perangkat PC saat pengujian berlangsung. Instalasi dapat dilakukan dengan menginstal Node.js pada laman <https://nodejs.org/en>. Versi Node.js yang digunakan pada penelitian ini adalah versi 8. Setelah Node.js terinstal maka dilanjutkan dengan instalasi Node-RED dengan memberi perintah pada Windows CMD sebagai berikut :

```
“npm install -g –unsafe-perm node-red”
```

```
“npm i node-red-contrib-midi”
```

Node-RED dapat dijalankan dengan memasukkan perintah “node-red” pada Windows CMD. Program Node-RED berjalan pada peladen lokal. Untuk mengakses panel Node-RED, dilakukan dengan peramban web pada alamat <http://localhost:1880>.

3. Instalasi dan Konfigurasi loopMIDI

Proses instalasi pada Windows dapat dilakukan dengan mengunduh dan menjalankan berkas instalasi pada laman <http://www.tobias-erichsen.de/software/>. Konfigurasi loopMIDI dengan menambahkan virtual MIDI bernama “loop MIDI” melalui *icon* “+”.

4. Instalasi dan Konfigurasi Poket MIDI

Pocket MIDI berfungsi sebagai MIDI *mapper* dan monitor. Proses instalasi pada Windows dilakukan dengan mengunduh dan menjalankan berkas instalasi pada laman <https://morson.jp/pocketmidi-webpage>. Konfigurasi dengan memilih masukan MIDI pada menu *InputPort* dengan “loop MIDI” dan keluaran MIDI pada menu *OutputPort* dengan USB-MIDI. Untuk melewati masukan MIDI ke keluaran MIDI dengan menuju menu “MIDI Settings” pada menu “View” dan mencentang “Thru” pada *port* yang disediakan.

3.4.2 Program Pada Modul Pengirim

Program ini ditulis dengan bahasa C++ dan terdiri dari 2 berkas utama yaitu *midi_type.h* dan *main.cpp*. Berkas *midi_type.h* berisi *header*, deklarasi variabel, dan *enum* byte MIDI yang digunakan dalam program utama pada *main.cpp*. Isi berkas *midi_type.h* dapat dilihat pada lampiran **halaman 42**. Berkas *main.cpp* berisi program utama dan fungsi-fungsi yang dipanggil dalam program utama. *Framework* Arduino yang digunakan memudahkan dalam penulisan program.

Sesuai pada **Gambar 3.7** diagram alur pengirim dan *Framework* Arduino, maka program terbagi menjadi 2 fungsi utama yaitu *setup()* dan *loop()*. *Setup* berisi inisialisasi sistem berupa inisialisasi *baudrate* serial, penyambungan Wi-Fi, dan inisialisasi IP dan *port* UDP. *Loop* adalah fungsi yang dipanggil berulang-ulang. *Loop* berisi proses pembacaan serial, identifikasi pesan dan pengiriman paket UDP. Penjelasan dan masing-masing fungsi dapat dilihat pada komentar lampiran **halaman 43**.

3.4.3 Program Pada Modul Penerima

Program menggunakan bahasa pemrograman C++ dan terdiri dari 1 berkas utama yaitu *main.cpp*. Terdapat 2 fungsi pada berkas *main.cpp* yaitu *setup* dan *loop*. *Setup* berisi inisialisasi sistem berupa inisialisasi *baudrate* serial, penyambungan Wi-Fi, dan fungsi *listening port* UDP. Pada *listening port*, apabila terdapat paket masuk maka isi paket akan ditulis ke serial MIDI. Fungsi *loop* tidak melakukan apa-apa.

3.5 Pengujian dan Analisis Sistem

Pada tahapan pengujian sistem, parameter yang dilakukan disesuaikan dengan analisa bagian dan keseluruhan fungsional sistem.

3.5.1 Pengujian Sistem Membaca Data MIDI

Pengujian dilakukan pada bagian pengirim sistem. Pengujian ini menguji bagian rangkaian pengondisi sinyal MIDI logika 5V ke 3,3V beserta fungsi ESP32 untuk membaca sinyal serial.

Proses pelaksanaan pengujian adalah dengan menghubungkan sinyal MIDI ke rangkaian pengondisi sinyal dan keluaran rangkaian pengondisi sinyal dihubungkan ke jalur serial pada ESP32 serta disediakan jalur serial keluaran yang dihubungkan ke komputer melalui USB-Serial. ESP32 diprogram untuk membaca data serial yang diterima melalui rangkaian

pengondisi sinyal dan memfilter pesan MIDI kemudian meneruskannya ke komputer melalui USB-Serial.

Proses menganalisis pengujian ini adalah dengan merekam data yang didapatkan dari ESP32 melalui USB-Serial ke dalam tabel dan mencocokkan data MIDI yang diterima dari ESP32 dengan data MIDI yang dikirimkan ke pengondisi sinyal. Kesimpulan yang didapatkan dari analisis pengujian ini adalah apabila data yang dikirim dan diterima bernilai sama, maka sistem bekerja dengan baik. Sebaliknya apabila data yang dikirim tidak sama dengan data yang direkam, maka sistem tidak bekerja dengan semestinya.

3.5.2 Pengujian Komunikasi UDP

Pengujian komunikasi UDP antara bagian pengirim dengan bagian penerima. Pengujian ini bertujuan untuk memastikan fungsi Wi-Fi beserta pengiriman paket UDP dapat bekerja pada sistem.

Pelaksanaan pengujian ini dengan memprogram bagian pengirim dan penerima supaya berfungsi sebagai Wi-Fi *Client* dan terhubung ke *Access Point*. Bagian pengirim akan mengirimkan data UDP secara *broadcast* ke bagian penerima. Bagian pengirim dan penerima terhubung ke komputer melalui sambungan serial dan mengirim log kegiatan yang telah terjadi.

Proses menganalisis pengujian ini adalah dengan melihat dan membandingkan log kegiatan serial pengirim dan penerima. Kesimpulan yang didapatkan adalah apabila jumlah data yang diterima bagian penerima sama dengan jumlah data yang dikirim bagian pengirim, maka sistem berjalan dengan baik. Dan apabila jumlah data yang diterima bagian penerima tidak sama dengan jumlah data yang dikirim bagian pengirim, maka sistem tidak berjalan dengan baik.

3.5.3 Pengujian Waktu Tunda Sistem

Pengujian bertujuan untuk mengukur jeda yang muncul akibat sistem. Nilai waktu tunda atau jeda ini muncul akibat dari perubahan dan pembungkusan data ke dalam bentuk baru. Pengujian ini dilakukan dengan mengukur jeda satu perangkat pengirim menuju satu perangkat penerima dan satu perangkat pengirim menuju dua perangkat penerima.

Proses pengujian adalah dengan menyusun sistem menjadi satu kesatuan yang sudah jadi. Pada bagian masukan MIDI perangkat penerima dihubungkan dengan sumber data MIDI OUT dan masukan *trigger* perangkat *Delay Measurement System* (DMS). Sedangkan keluaran MIDI perangkat penerima dihubungkan dengan masukan perangkat DMS.

DMS akan mengukur jeda dan mengirimkan ke komputer melalui jalur serial. Kesimpulan yang didapatkan dari pengujian ini adalah nilai data yang dilewatkan ke sistem beserta nilai waktu tunggu berupa rata-rata, maksimal, dan minimal sistem.

3.6 Analisa Hasil dan Kesimpulan

Analisa hasil dan kesimpulan pada penelitian ini didapatkan apabila seluruh pengujian sistem telah selesai dilakukan. Kesimpulan dihasilkan dari hasil pengujian dan analisa penelitian.

BAB IV HASIL DAN PEMBAHASAN

Hasil dan pembahasan dalam penelitian ini dilakukan untuk mengetahui kinerja seluruh sistem. Pengujian yang dilakukan merupakan pengujian fungsional. Pengujian dilakukan pada fungsi-fungsi dasar sistem. Pengujian yang dilakukan adalah sebagai berikut:

1. Pengujian Sistem Membaca Data MIDI
2. Pengujian Komunikasi UDP
3. Pengujian Waktu Tunda Sistem

4.1 Pengujian Sistem Membaca Data MIDI

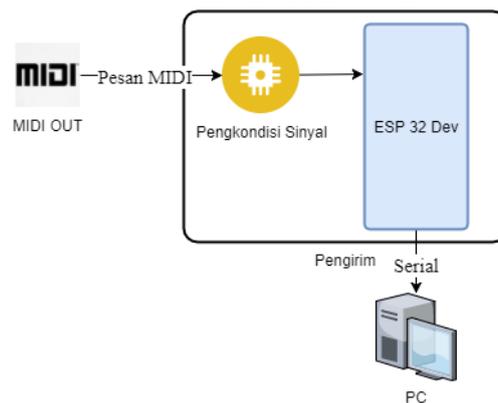
Tujuan pengujian adalah memastikan rangkaian pengondisi sinyal dan proses identifikasi pesan MIDI pada serial bekerja dengan baik. Proses pengujian ini menguji identifikasi 16 pesan MIDI. Pesan MIDI akan diidentifikasi jumlah bit dan data di dalamnya.

Alat yang digunakan pada pengujian membaca data MIDI sebagai berikut:

1. PC Windows
2. Modul pengirim
3. Kabel USB

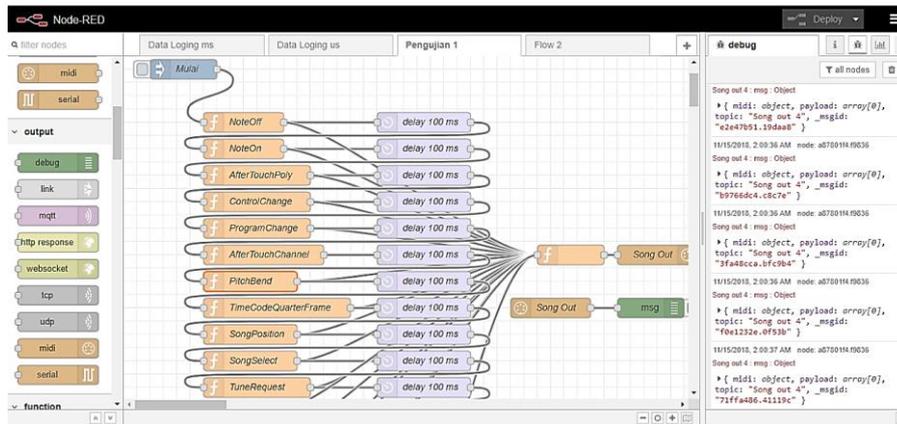
4.1.1 Prosedur Percobaan

Pengujian berlangsung dengan merangkai alat sesuai **Gambar 4.1** dan memprogram pengirim untuk membaca semua pesan MIDI yang masuk, mengidentifikasi pesan MIDI tersebut, dan menyampaikan hasil identifikasi ke PC melalui jalur serial. Program pengujian dapat dilihat pada lampiran **halaman 49**.



Gambar 4.1 Rangkaian Pengujian Sistem Membaca Data MIDI

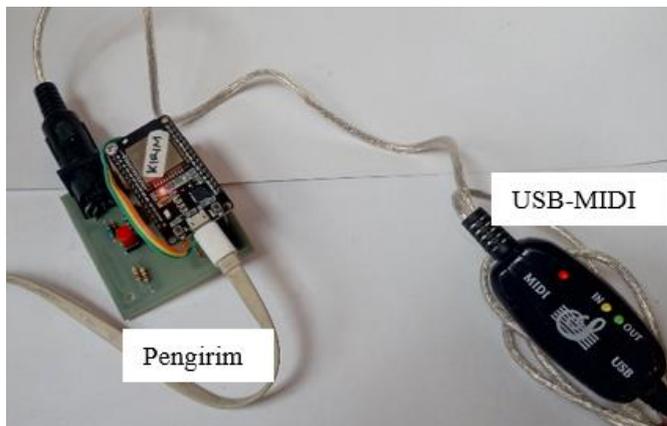
Sumber pesan MIDI adalah PC melalui USB-MIDI. Program yang digunakan untuk mensimulasikan pesan MIDI adalah Node-RED, loopMIDI dan Pocket MIDI. Node-RED diprogram untuk mengirimkan 16 jenis pesan MIDI dengan jeda antara pesan 100 ms menuju port “loop MIDI”. Buka Pocket MIDI dan pilih *inputPort* “loop MIDI” dan *outputPort* “USB-MIDI”. **Gambar 4.2** adalah tangkapan layar program Node-RED.



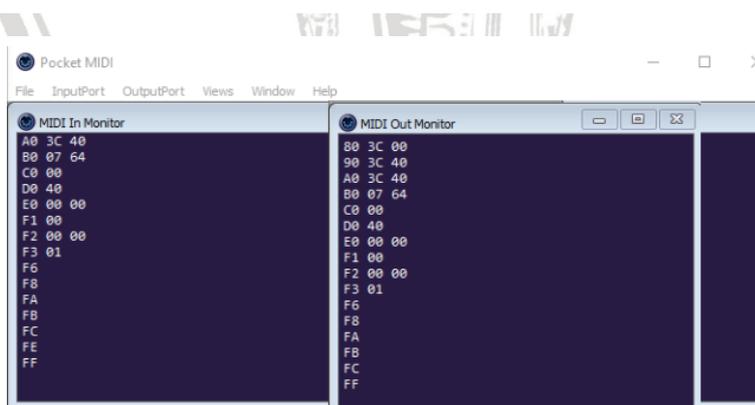
Gambar 4.2 Tangkapan Layar Aplikasi Node-RED

4.1.2 Hasil Pengujian dan Analisis

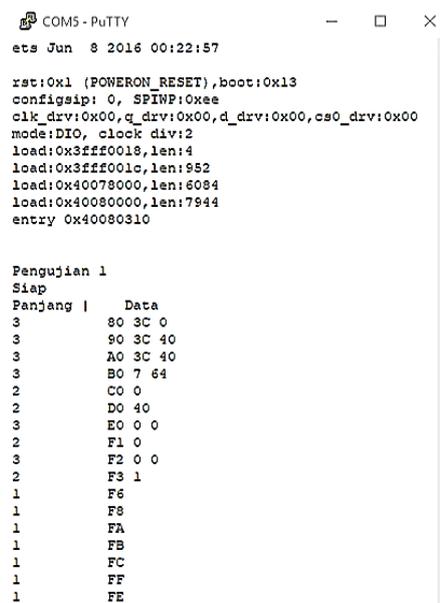
Berikut ini adalah gambar ketika pengujian berlangsung.



Gambar 4.3 Pengujian Sistem Membaca Data MIDI



Gambar 4.4 Tangkapan Layar Keluaran MIDI pada Pocket MIDI



```

COM5 - PuTTY
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13
configsiip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00
mode:DIO, clock div:2
load:0x3fff0018,len:4
load:0x3fff001c,len:952
load:0x40078000,len:6084
load:0x40080000,len:7844
entry 0x40080310

Pengujian 1
Siap
Panjang | Data
3      | 80 3C 0
3      | 90 3C 40
3      | A0 3C 40
3      | B0 7 64
2      | C0 0
2      | D0 40
3      | E0 0 0
2      | F1 0
3      | F2 0 0
2      | F3 1
1      | F6
1      | F8
1      | FA
1      | FB
1      | FC
1      | FE

```

Gambar 4.5 Tangkapan Layar Serial Pengujian Membaca Data MIDI

Tabel 4.1 Pengujian Sistem Membaca Data MIDI

<i>Pesan</i>	Data (hex)	Besar (byte)	Terbaca
<i>NoteOff</i>	80,3C,0	3	Terbaca
<i>NoteOn</i>	90,3C,40	3	Terbaca
<i>AffterTouchPoly</i>	A0,3C,40	3	Terbaca
<i>ControlChange</i>	B0,7,64	3	Terbaca
<i>ProgramChange</i>	C0,0	2	Terbaca
<i>AffterTouchChannel</i>	D0,40	2	Terbaca
<i>PitchBend</i>	E0,0,0	3	Terbaca
<i>TimeCodeQuarterFrame</i>	F1,0	2	Terbaca
<i>SongPotition</i>	F2,0,0	3	Terbaca
<i>SongSelect</i>	F3,1	2	Terbaca
<i>TuneRequest</i>	F6	1	Terbaca
<i>Clock</i>	F8	1	Terbaca
<i>Start</i>	FA	1	Terbaca
<i>Continue</i>	FB	1	Terbaca
<i>Stop</i>	FC	1	Terbaca
<i>ActiveSensing</i>	FE	1	Terbaca
<i>SystemReset</i>	FF	1	Terbaca

Dari pengujian, didapatkan bahwa semua pesan yang dikirimkan dapat teridentifikasi dengan baik. Hal tersebut terlihat dari pesan serial yang ditampilkan sama dengan pesan yang dikirim.

4.2 Pengujian Komunikasi UDP

Pengujian dilakukan untuk memastikan perangkat dapat berkomunikasi dengan kecepatan sesuai spesifikasi MIDI. *Baudrate* yang digunakan MIDI DIN adalah 31,250

kbaud, apabila diubah menjadi kbps maka setara dengan 3,125kbps. Pengujian ini menggunakan acuan kecepatan lebih dari *baudrate* MIDI yaitu 16kbps.

Alat yang digunakan pada pengujian komunikasi UDP sebagai berikut:

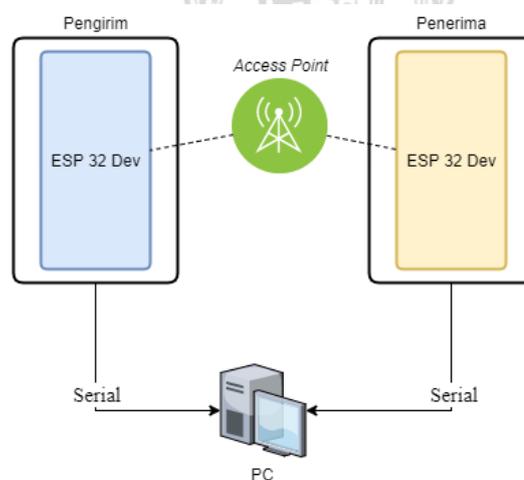
1. PC Windows
2. *Access Point*
3. Modul Pengirim
4. Modul Penerima
5. Kabel USB

4.2.1 Prosedur Percobaan

Pengujian dilakukan dengan memprogram perangkat pengirim dan penerima supaya terhubung pada *Access Point* yang sama dan perangkat pengirim mengirimkan pesan UDP *Broadcast* sebesar 64 byte sebanyak 100 kali dengan jeda di antara pengiriman paket sebesar 4 ms.

Perangkat pengirim diprogram untuk terhubung ke SSID bernama “MIDI” dan mulai mengirim paket UDP sebanyak 64 bit setiap 4 ms. Alamat UDP adalah *broadcast* “255.255.255.255” dengan *port* 1112. Isi satu paket UDP adalah karakter ASCII ‘A’ hingga ‘Z’ sebanyak 64 kali dengan paket pertama berisi “A”, paket ke-dua berisi “B”, dan seterusnya. Pengaturan jeda 4 ms diatur melalui fungsi *Ticker* yang disediakan ESP32 sehingga lebih akurat. Setelah 100 paket UDP terkirim, pengirim akan melaporkan jumlah dan besar semua paket yang sudah dikirim ke PC melalui jalur serial.

Perangkat penerima diprogram untuk terhubung ke SSID bernama “MIDI” dan mulai mendengarkan paket UDP pada port 1112. Pesan yang masuk dihitung banyaknya dan besar semua paket yang diterima. Proses pelaporan paket yang diterima penerima melalui mekanisme tombol pada pin D23. Apabila pin D23 berlogika *low* maka penerima akan melaporkan jumlah dan besar semua paket yang dia terima ke PC melalui jalur serial. Untuk menampilkan Program pengirim dan penerima dapat dilihat pada lampiran **halaman 53** dan **halaman 55**.

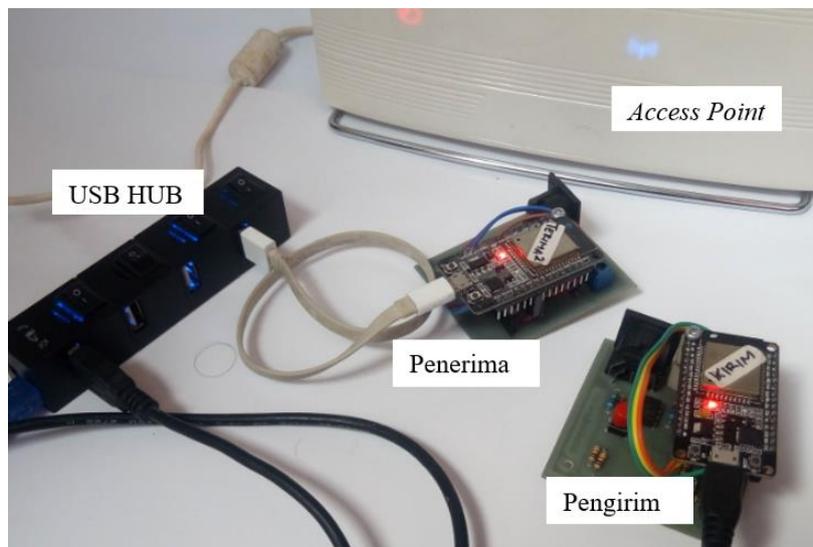


Gambar 4.6 Rangkaian Pengujian Komunikasi UDP

Perangkat dirangkai sesuai **Gambar 4.6**. Dan percobaan siap dimulai dengan menekan tombol boot pada ke dua modul dan melepaskan tombol secara bersamaan.

4.2.2 Hasil Pengujian dan Analisis

Berikut ini adalah gambar ketika pengujian berlangsung.



Gambar 4.7 Pengujian Komunikasi UDP

<pre> COM5 - PuTTY ets Jun 8 2016 00:22:57 rst:0x1 (POWERON RESET),boot:0x13 (SPI_FAST_FLASH_BOOT) configspi: 0, SPIWP:0xee clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00 mode:DIO, clock div:2 load:0x3fff0018,len:4 load:0x3fff001c,len:952 load:0x40078000,len:6084 load:0x40080000,len:7944 entry 0x40080310 Pengujian 2 UDP Async Send data I'M OK ms : 4 UDP connected : 255, 255,255, 255 ===== Paket Terkirim = 100 Paket Sebesar= 6400 </pre> <p style="text-align: center;">PENGIRIM</p>	<pre> COM19 - PuTTY ets_main.c 371 ets Jun 8 2016 00:22:57 rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT) configspi: 0, SPIWP:0xee clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00 mode:DIO, clock div:2 load:0x3fff0018,len:4 load:0x3fff001c,len:952 load:0x40078000,len:6084 load:0x40080000,len:7944 entry 0x40080310 Menghubungkan ke AP: MIDI UDP Listening on IP: 192.168.0.194 Port: 1112 ===== Paket Diterima= 100 Paket Sebesar= 6400 </pre> <p style="text-align: center;">PENERIMA</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 4.8 Tangkapan Layar Serial Pengujian Komunikasi UDP

Dari pengujian didapatkan bahwa perangkat pengirim mengirim paket UDP sebesar 64 byte sebanyak 100 kali dan penerima menerima paket UDP sebesar 64 byte sebanyak 100 kali. Sehingga paket yang dikirim pengirim diterima semua oleh penerima.

4.3 Pengujian Waktu Tunda Sistem

Pengujian waktu tunda sistem bertujuan untuk mengetahui jeda yang ditimbulkan akibat proses pembungkusan dan pengiriman pesan MIDI melalui sistem. Pengujian ini meliputi pengujian jeda yang timbul pada satu pengirim ke satu penerima dan satu pengirim ke dua penerima.

Alat yang digunakan pada pengujian waktu tunda sistem sebagai berikut:

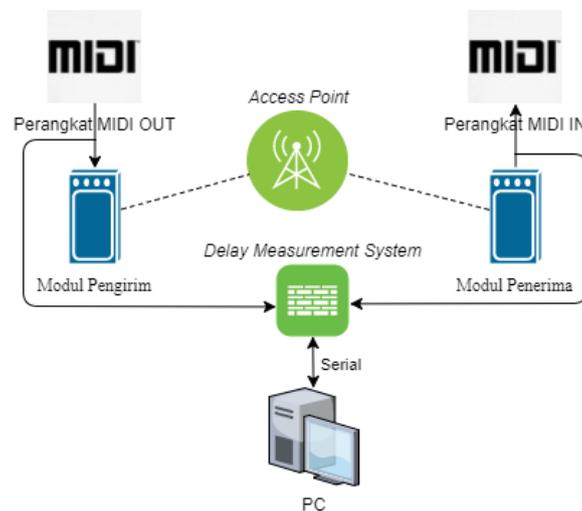
1. PC Windows
2. *Access Point*
3. Modul Pengirim
4. Modul Penerima
5. Kabel USB
6. Arduino Mega 2560
7. Kabel konektor

4.3.1 Prosedur Percobaan

Pengujian ini dilaksanakan dengan menghubungkan perangkat ke access point dan mengirimkan satu pesan MIDI ke perangkat pengirim setiap 500 ms sebanyak 1000 kali serta mengukur waktu tunda yang muncul. Pesan MIDI yang digunakan adalah *ActiveSensing* (0xfe) melalui aplikasi Pocket MIDI pada PC.

Pocket MIDI diatur untuk mengeluarkan pesan *ActiveSensing* dengan memilih menu *View*→*Active Sensing*. Pengaturan waktu *Active Sensing* diubah ke 500 ms. Keluaran MIDI dipilih USB-MIDI. Konektor MIDI OUT pada USB-MIDI dihubungkan ke perangkat pengirim. Pengujian siap untuk dimulai.

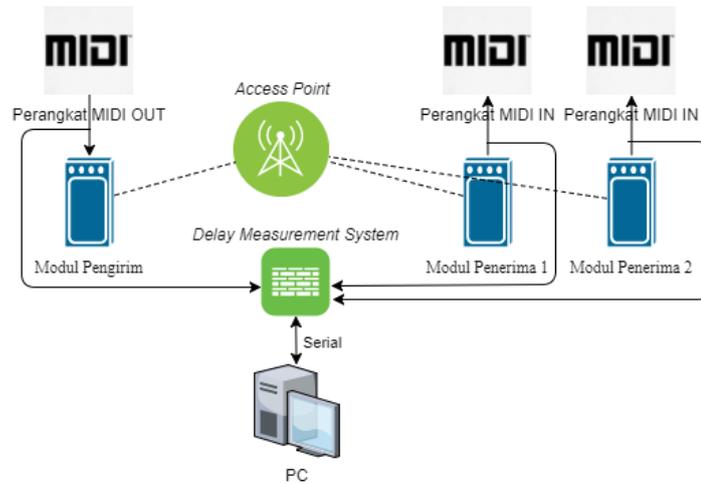
Waktu tunda didapatkan dengan mengukur jeda yang timbul dari sinyal masuk ke perangkat pengirim hingga sinyal keluar di perangkat penerima. Pengukuran dilakukan dengan menggunakan *Delay Measurement System* berupa waktu tunggu antara *interrupt* mikrokontroler Arduino Mega 2560.



Gambar 4.9 Rangkaian Pengujian Waktu Tunda Satu Penerima

Pengukuran jeda pertama adalah dengan mengukur jeda yang timbul pada satu pengirim ke satu penerima. Diagram pengujian dapat dilihat pada **Gambar 4.9**. DMS (Arduino Mega) diprogram untuk mendeteksi penurunan logika tegangan pada masukan perangkat pengirim dan keluaran logika tegangan perangkat penerima. Selisih perubahan antara dua pin dihitung dan dikirim ke PC melalui serial. Pin yang digunakan pada Arduino Mega adalah pin D18 sebagai *interrupt* pengirim dan pin D2 sebagai *interrupt* penerima. Pin D3 diberi tegangan

high karena hanya diperlukan 2 *interrupt* dan digunakan pada pengukuran dua penerima. Program Arduino Mega dapat dilihat pada lampiran **halaman 57**.

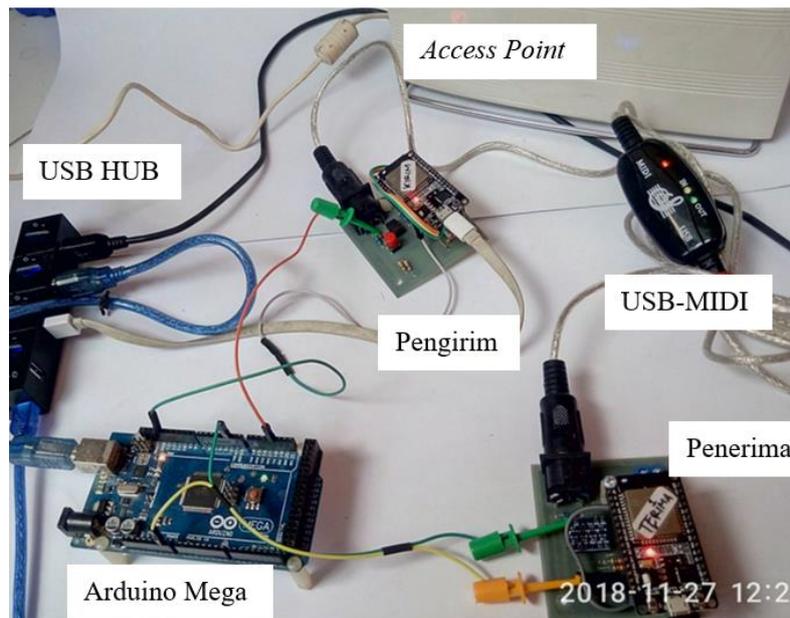


Gambar 4.10 Rangkaian Pengujian Waktu Tunda Dua Penerima

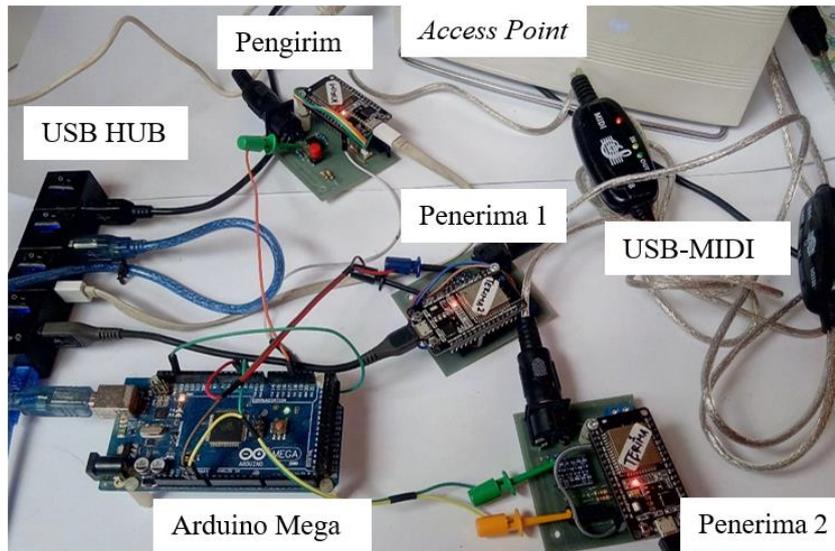
Pengukuran jeda kedua adalah dengan mengukur jeda yang timbul pada satu pengirim ke dua penerima. Diagram pengujian dapat dilihat pada **Gambar 4.10**. DMS yang digunakan adalah Arduino Mega dengan konfigurasi pin Arduino Mega D18 sebagai *interrupt* pengirim, pin D2 sebagai *interrupt* penerima1 dan D3 sebagai *interrupt* penerima 2. PC akan merekam data dalam bentuk serial log dan dilanjutkan dengan mengolah data tersebut menggunakan aplikasi *spreedsheet*.

4.3.2 Hasil Pengujian dan Analisis

Pengujian jeda satu penerima menghasilkan 1000 data jeda dan dua penerima dengan 2000 data jeda dengan persebaran yang acak. Data diolah dan dikelompokkan berdasarkan jeda. Berikut ini adalah hasil pengujian.



Gambar 4.11 Perangkat Pengujian Waktu Tunda Satu Penerima



Gambar 4.12 Perangkat Pengujian Waktu Tunda Dua Penerima

Pengujian waktu tunda sistem apabila perangkat penerima terdiri dari 1 penerima.

Tabel 4.2 Pengujian Waktu Tunda Sistem dengan Satu Penerima

<i>Jeda (ms)</i>	<i>Cacah Pesan</i>
1	0
2	432
3	514
4	45
5	4
6	3
7	1
8	1
9	0
<u>Jumlah</u>	<u>1000</u>

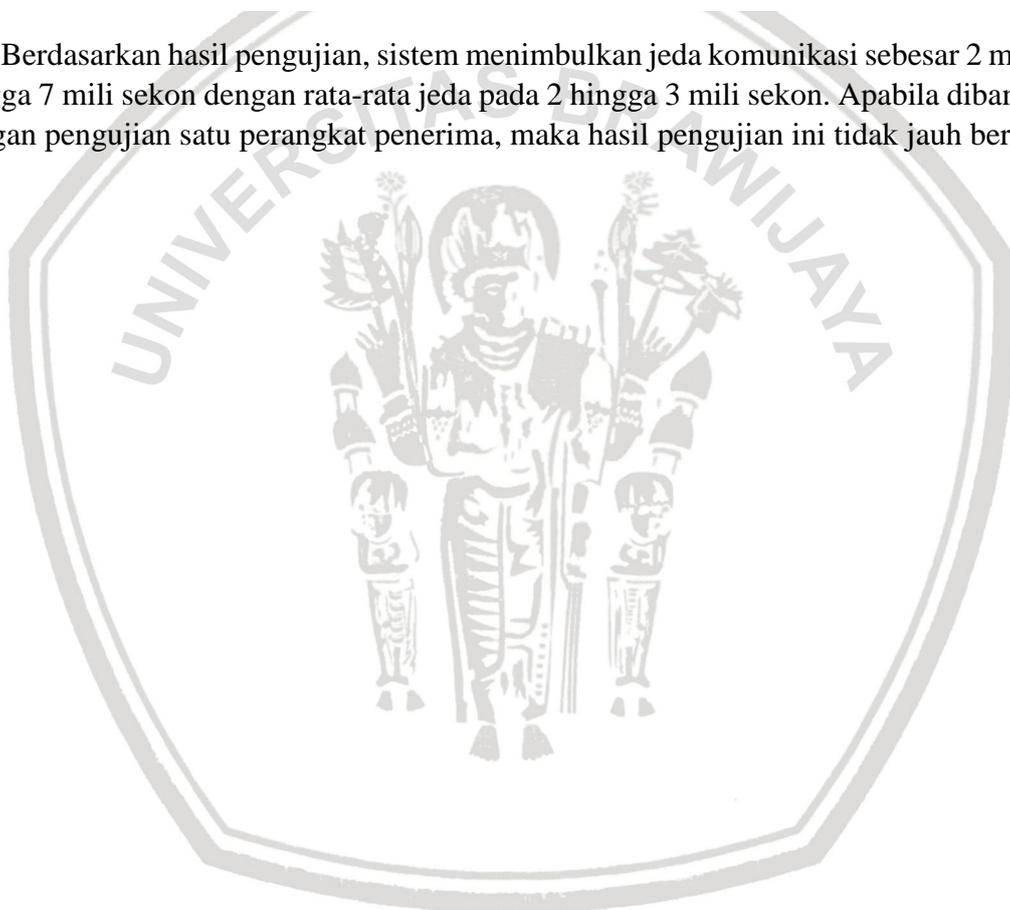
Hasil pengujian menunjukkan bahwa sistem menimbulkan jeda pengiriman sebesar 2 mili sekon hingga 8 mili sekon dengan rata-rata jeda pada 2 hingga 3 mili sekon. Hasil pengujian ini akan dibandingkan dengan pengujian menggunakan dua penerima.

Pengujian waktu tunda sistem apabila perangkat penerima terdiri dari 2 penerima.

Tabel 4.2 Pengujian Waktu Tunda Sistem dengan Dua Penerima

<i>Jeda (ms)</i>	<i>Cacah Pesan</i>	
	Penerima 1	Penerima 2
1	0	0
2	431	419
3	505	516
4	50	51
5	7	8
6	6	5
7	1	1
8	0	0
9	0	0
<i>Jumlah</i>	<u>1000</u>	<u>1000</u>

Berdasarkan hasil pengujian, sistem menimbulkan jeda komunikasi sebesar 2 mili sekon hingga 7 mili sekon dengan rata-rata jeda pada 2 hingga 3 mili sekon. Apabila dibandingkan dengan pengujian satu perangkat penerima, maka hasil pengujian ini tidak jauh berbeda.



BAB V KESIMPULAN DAN SARAN

Bab ini memuat penarikan kesimpulan dan saran yang diharapkan dapat berguna untuk pengembangan selanjutnya berdasarkan tahapan yang telah dilakukan sebelumnya dari penelitian ini.

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang sudah dilaksanakan, maka dapat diambil kesimpulan bahwa :

1. ESP32 dapat digunakan sebagai perantara komunikasi MIDI menggunakan jaringan Wi-Fi dengan protokol komunikasi UDP secara *broadcast*.
2. Rancangan dan implementasi komunikasi MIDI melalui protokol UDP Broadcast adalah dengan membungkus satu atau banyak pesan MIDI menjadi 1 paket UDP dan mengirimkannya ke alamat *broadcast* jaringan. Setelah sampai tujuan, paket UDP dibuka dan dikembalikan menjadi pesan MIDI.
3. Jeda yang muncul di antara pengirim dan penerima tergolong relatif kecil dan tidak terjadi perbedaan signifikan apabila antara 1 dan 2 perangkat penerima.

5.2 Saran

Saran yang dapat digunakan dalam peningkatan kerja sistem dapat diuraikan sebagai berikut:

1. Komunikasi MIDI menggunakan protokol yang aman dari data yang hilang.
2. Memanfaatkan 2 *core* ESP32 untuk pengolahan dan komunikasi.
3. Penanggulangan dan koreksi kesalahan pada pesan yang hilang.
4. Komunikasi MIDI menggunakan jenis komunikasi lainnya.
5. Manajemen tugas pada *Real Time Operational System* (RTOS).

DAFTAR PUSTAKA

- Bartolomeu, P. J. (2005). *Evaluating bluetooth for the wireless transmission of MIDI*. Aveiro: Universidade de Aveiro.
- IEEE. (2016, Desember 14). *IEEE 802.11 Standard*. Diambil 12 September 2018 dari Standard IEEE: https://standards.ieee.org/standard/802_11-2016.html
- Kurose, J. F., & Ross, K. W. (2013). *COMPUTER NETWORKING A Top-Down Approach SIXTH EDITION*. Boston: Pearson.
- Lehrman, P. D., & Tully, T. (2017). *What is MIDI?* Diambil 12 September 2018 dari "What is MIDI" Guide by Paul Lehrman: <https://www.midi.org/articles-old/what-is-midi-guide-by-paul-lehrman>
- Murray, D. (2015, Juni 17). *Farewell R.I.P. Firewire, Hello USB!*. Diambil 12 September 2018 dari Youtube: <https://www.youtube.com/watch?v=rJ5bao4o3pk>
- Postel, J. (1980, August 28). *RFC 768 - User Datagram Protocol*. Diambil 12 September 2018 dari IETF Tools: <https://tools.ietf.org/html/rfc768>
- Schutte, H. (1997). *Bi-directional level shifter for PC-bus and other systems*. Application Note AN97055. The Netherlands: Philips Semiconductors Systems Laboratory Eindhoven.
- The MIDI Manufacturers Association. (2014). *The Complete MIDI 1.0 Detailed Specification Incorporating all Recommended Practices document version 96.1 third edition*. Los Angeles: PDF Presentation.
- Young, J. P., & Fujinaga, I. (1999). *Piano Master Classes Via The Internet*. ResearchGate, 2508134.