JIRK
Journal of Innovation Research and Knowledge
Vol.3, No.8, Januari 2024

IMPLEMENTASI MQTT PADA SISTEM KENDALI NIRKABEL ROBOT KAMERAMAN BERBASIS MIKROKONTROLER

Oleh
Juprianus Rusman[1], Gidion A. N. Pongdatu[2], Samrius Upa'[3]
[1,2,3]Teknik Informatika, Universitas Kristen Indonesia Toraja, Tana Toraja, Sulawesi Selatan, Indonesia
E-mail: [1]rusman.jr@ukitoraja.ac.id, [2]dionpongdatu@ukitoraja.ac.id, [3]samrius@ukitoraja.ac.id

Article History:
Received: 21-12-2023
Revised: 29-12-2023
Accepted: 24-01-2024

Keywords:
Kontrol Nirkabel, Protokol-MQTT, Latensi, Mikrokontroler

Abstract: *Dokumentasi berbasis video dari kegiatan dapat digunakan sebagai bukti efektivitas kegiatan serta media promosi. Kamerawan yang bertugas merekam kegiatan sering kali menghadapi hambatan fisik saat memindahkan kabel komunikasi kamera dengan perangkat video switcher yang mengganggu proses dokumentasi. Sebagai solusi, dapat menggunakan robot kamerawan dengan kontrol nirkabel, namun pengembangannya memiliki tantangan dalam efisiensi komunikasi antara robot dan perangkat pengontrol. Komunikasi yang tidak efisien dapat menyebabkan keterlambatan dalam mengendalikan robot, keterlambatan dalam pengiriman data, atau bahkan kehilangan data. Oleh karena itu, mekanisme komunikasi yang efisien diperlukan untuk mengatasi tantangan ini. Dalam penelitian ini, protokol Message Queuing Telemetry Transport (MQTT) diterapkan sebagai sistem kontrol nirkabel untuk robot kamerawan berbasis mikrokontroler. Implementasi protokol MQTT pada mikrokontroler ESP8266 sebagai pengontrol (publisher), mikrokontroler ESP32 pada robot (subscriber), dan raspberry sebagai broker. Evaluasi latensi dengan metode roundtrip time (RTT) menunjukkan waktu rata-rata sebesar 0,7338 ms antara pengontrol dan broker, serta 1,68045 ms antara robot kamerawan dan broker. Hasil ini menunjukkan bahwa protokol MQTT mampu menjadi alternatif transmisi data yang efisien untuk mengendalikan robot kamerawan secara nirkabel.*

PENDAHULUAN

Universitas Kristen Indonesia Toraja merupakan salah satu perguruan tinggi swasta yang berada di Kabupaten Tana Toraja dan Kabupaten Toraja Utara Provinsi Sulawesi Selatan. Dalam pelaksanaan kegiatan seperti wisuda, dilakukan *live streaming* video melalui kanal *youtube* agar sanak keluarga dari wisudawan yang tidak dapat memasuki ruang wisuda atau yang berada di rumah dapat melihat prosesi wisuda secara daring. Kameraman yang bertindak sebagai pembawa kamera rekaman memiliki tugas untuk mendapatkan rekaman saat berlangsungnya kegiatan. Namun dalam melaksanakan tugas, kameraman sering

melintas atau lalu lalang dengan memindahkan atau menarik kabel komunikasi kamera dengan perangkat video *switcher*. Hal ini tentu merepotkan bagi kameraman dan dapat menggangu kegiatan saat berlangsung.

Pada era digital saat ini, tugas seorang kameraman dapat diganti dengan robot kameraman. Robot kameraman dengan kendali nirkabel menjadi solusi yang populer dalam dokumentasi atau pengambilan gambar dan video yang dinamis dan fleksibel. Namun dalam penerapannya, salah satu tantangan dalam pengembangan robot kameraman nirkabel adalah efisiensi komunikasi antara robot dan perangkat pengendali. Komunikasi yang tidak efisien dapat mengakibatkan keterlambatan dalam mengontrol robot, penundaan dalam pengiriman data, atau bahkan kehilangan data, yang dapat mempengaruhi kualitas pengambilan gambar atau video. Oleh karena itu, diperlukan suatu mekanisme komunikasi yang efisien, handal, dan ringan untuk mengatasi tantangan tersebut. *Message Queuing Telemetry Transport* (MQTT) adalah salah satu protokol komunikasi yang dapat menjadi solusi untuk tantangan tersebut. MQTT adalah protokol komunikasi berbasis pesan antar mesin dengan mesin (M2M) yang ringan, handal, dan hemat energi, yang dirancang untuk aplikasi *Internet of Things* (IoT) [1]–[3]. Dalam konteks penelitian ini, protokol MQTT digunakan untuk komunikasi antara robot kameraman dan perangkat pengendali secara nirkabel. Mikrokontroler adalah salah satu platform perangkat keras yang digunakan dalam pengembangan sistem kendali nirkabel, karena ukurannya yang kecil, konsumsi energi yang rendah, dan kemampuan pemrosesan data secara real-time [4].

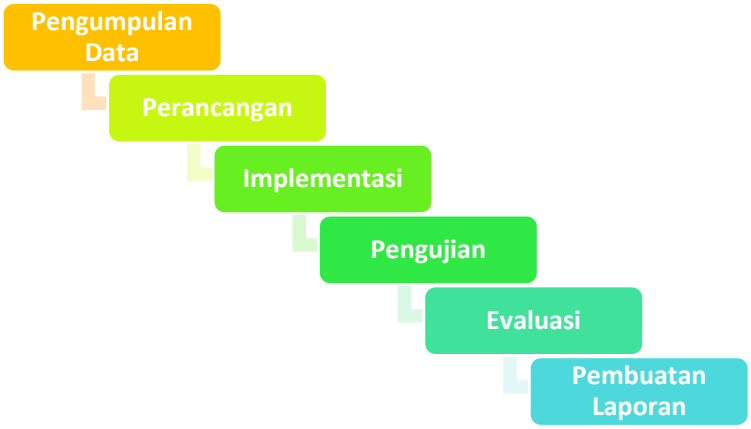
Implementasi yang terkait telah dilakukan sebelumnya seperti pegembangan sistem kendali robot industri untuk pengelasan [5]. Penelitian ini menghasilkan pengontrol generik dalam bentuk blok fungsi/prosedur sebagai basis pengembangan untuk aplikasi kontrol pengelasan akhir. Penelitian selanjutnya yaitu pengontrolan terhadap robot rescue dengan menggunakan komunikasi Zigbee dengan hasil jarak komunikasi efektif antara robot rescue dan pengontrol jarak jauh adalah 120 m di luar ruangan [6]. Penelitian yang serupa yaitu penelitian untuk pengembangan pengontrolan robot lengan menggunakan mikrokontroler Arduino dengan komunikasi Bluetooth [4] dengan hasil kesalahan pergerakan robot untuk sumbu x sekitar 1,38% dan kesalahan pergerakan robot untuk sumbu y 12,14%.

Penelitian ini berfokus pada mekanisme komunikasi yang efisien untuk pengendalian nirkabel robot kameraman menggunakan protokol MQTT. Metode/protokol MQTT diimplementasikan pada mikrokontroler pengendali *(trasmitter)* dan mikrokontroler penerima *(receiver)* pada robot kameraman. Media transmisi yang digunakan yaitu media transmisi nirkabel *(wireless).* Adapun perangkat pengendali yaitu *joystick* yang dihubungkan dengan mikrokontroler pengendali. Pada robot kameraman akan digunakan perintah untuk kendali akselerasi maju, mundur, putar kiri, putar kanan dan berhenti.

ISSN 2798-3471 (Cetak)
ISSN 2798-3641 (Online)

METODE PENELITIAN

Metode yang digunakan pada penelitian ini ditunjukkan pada gambar 1.



Gambar 1. Tahapan penelitian

Tahap awal penelitian dilakukan dengan pengumpulan data melalui observasi dan studi literatur. Observasi dilakukan untuk mengetahui bagaimana pengambilan gambar oleh kameraman. Studi literatur dilakukan untuk mengetahui metode yang digunakan dan penelitian yang telah dilakukan sebelumnya. Selanjutnya mengidentifikasi masalah berdasarkan latar belakang yang ada yaitu bagaimana mekanisme komunikasi yang efisien untuk pengendalian nirkabel robot kameraman menggunakan protokol MQTT.

Pada tahap perancangan dilakukan desain mekanisme/alur pengiriman dan penerimaan data menggunakan protokol MQTT dalam bentuk diagram alir *(flowchart).* Hasil pada tahapan ini yakni tersedianya desain perangkat lunak dan desain perangkat keras. Proses yang akan dilakukan pada tahap implementasi yaitu menterjemahkan desain perangkat lunak *(flowchat)* kedalam program *(sketch)* dengan stuktur bahasa pemrograman C yang kemudian akan ditanamkan kedalam mikrokontroler baik mikrokontroler pengirim/pengendali maupun mikrokontroler penerima pada robot kameraman.

Setelah perangkat lunak dan perangkat keras diintegrasikan, maka akan dilakukan pengujian untuk mengetahui unjuk kerja dari sistem kendali nirkabel yang telah dibangun. Pengujian dilakukan dengan pengendalian robot kameraman kemudian mengamati pergerakan robot dan menganalisis lalulintas jaringan dengan menggunakan *tools* Wireshark. Selanjutnya, hasil dari pengujian akan dievaluasi untuk mengetahui *quality of service* (QoS) dari jaringan yang dibangun. Tahap terakhir yaitu menuangkan hasil penelitian dalam bentuk lisan sebagai laporan penelitian atau pertanggungjawaban dan artikel publikasi.

HASIL DAN PEMBAHASAN

Hasil pelaksanaan penelitian dapat dijabarkan sebagai berikut.

Kebutuhan Perangkat Keras dan Perangkat Lunak

Langkah pertama pada penelitian adalah mendefenisikan komponen dan prinsip kerja dari sistem yang akan dibangun baik dari perangkat keras untuk infrastruktur jaringan, routing dan algoritma komunikasi data metode MQTT. Blok diagram sistem yang dibangun ditunjukkan pada gambar 2 berikut.

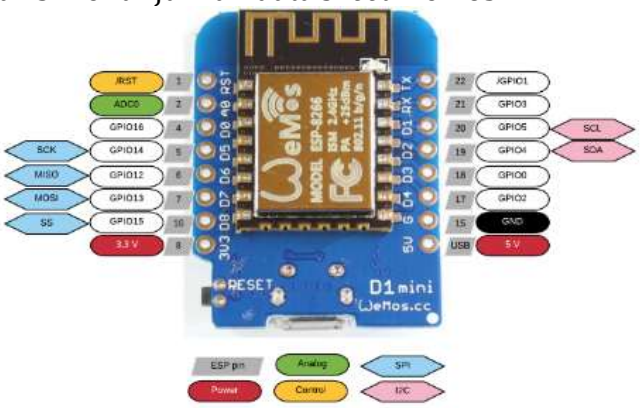
Gambar 2. Blok diagram sistem

Perangkat keras yang digunakan pada *publisher* yaitu *joystick* dan mikrokontoler. Kontroler *dualshock joystick* untuk memudahkan kendali dengan tombol yang memadai, sedangkan mikrokontroler yang digunakan yaitu ESP8266 dengan modul wemos D1 mini yang dapat melakukan komunikasi data secara nirkabel [7]. *Joystick* dengan Wemos D1 mini dihubungkan dengan metode *Serial Peripheral Interface* (SPI) yang merupakan sebuah protokol komunikasi serial untuk mentransfer data antara mikrokontroler (atau perangkat pengendali lainnya). SPI menggunakan metode sinkron, yang berarti komunikasi antara perangkat terjadi secara bersamaan dan diatur oleh sinyal *clock* yang bersamaan.

Pada broker digunakan modul *raspberry* yang dilengkapi dengan perangkat lunak *mosquitto*. Komponen ini akan bertindak sebagai *broker* sekaligus *gateway* antara *publisher* dan *subscriber*. Modul ini akan dilengkapi dengan *access point* sebagai alat bantu penguat daya pancar sinyal. Pada *subscriber* digunakan robot *mobile* dengan kendali utama mikrokontroler ESP32 dengan komponen pendukung lainnya seperti motor DC, driver motor L298N dan *power supply*. Perangkat lunak yang digunakan adalah Arduino IDE untuk membangun/implementasi algoritma dalam bentuk *sketch*/koding dengan struktur bahasa pemrograman C. Agar dapat memantau lalulintas jaringan maka digunakan aplikasi *wireshark*. Aplikasi ini dapat digunakan untuk monitoring komunikasi data dengan metode TCP/IP baik HTTP maupun MQTT dan beberapa metode lainnya.

Perancangan dan Implementasi Perangkat Keras

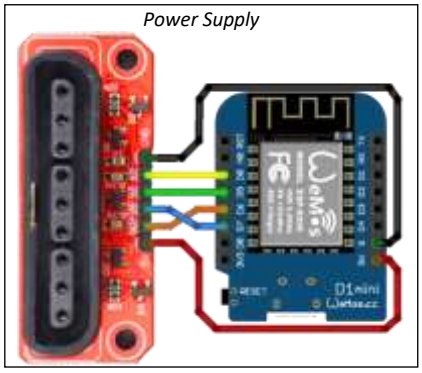
Perancangan modul pada bagian *publisher* meliputi komunikasi *Joystick* dengan Mikrokontroler menggunakan metode SPI. Metode SPI melibatkan komunikasi dengan menggunakan 4 *channel* (pin) antara lain *Serial Clock* (SCLK) yang digunakan untuk mengatur laju transfer data antara perangkat pengirim dan perangkat penerima. *Master In Slave Out* (MISO) digunakan oleh perangkat pengirim untuk mengirim data ke perangkat penerima. *Master Out Slave In* (MOSI) digunakan oleh perangkat pengirim untuk menerima data dari perangkat penerima dan *Slave Select/Chip Select* (SS/CS) adalah sinyal yang digunakan untuk memilih perangkat *peripheral* tertentu yang akan berkomunikasi dengan mikrokontroler. Gambar 3 menunjukkan data *sheet* wemos D1 mini.



Gambar 3. Data *sheet* wemos D1 mini

ISSN 2798-3471 (Cetak)
ISSN 2798-3641 (Online)

Selanjutnya mikrokontroler dihubungkan dengan adapter kontroler *joystick* dengan skematik seperti pada gambar 4.



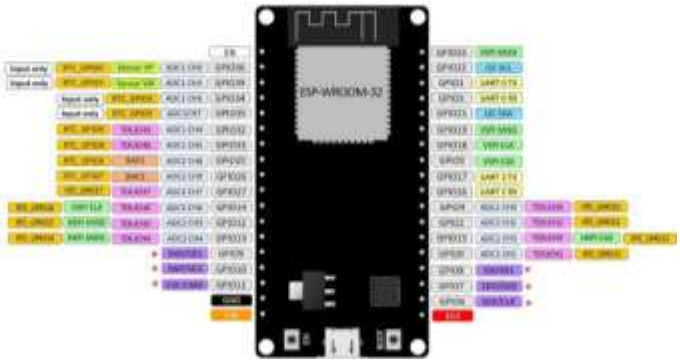
Gambar 4. Skematik komunikasi SPI wemos D1 mini dengan adapter *joystick*

Implementasi dari skematik pada gambar 4 sebagai modul *publisher* ditunjukkan pada gambar 5 berikut.

Gambar 5. Modul *publisher*

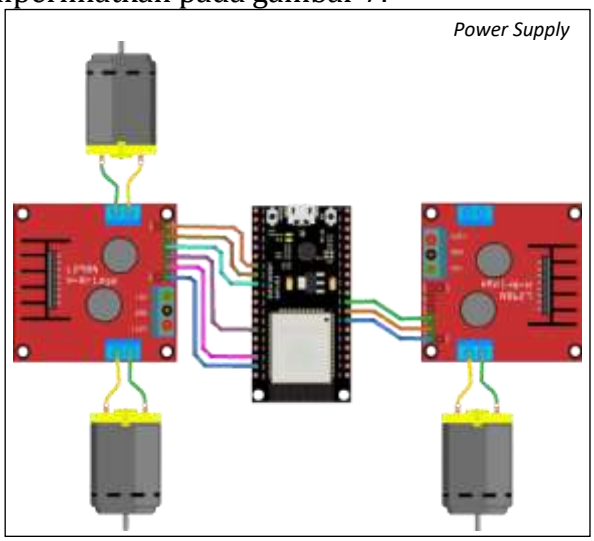
Mikrokontroler *publisher* ini selanjutnya akan melakukan komunikasi dengan *broker* menggunakan metode MQTT. Terlebih dahulu dilakukan konfigurasi TCP/IP karena protokol MQTT berjalan diatas *stack* TCP/IP. Konfigurasi TCP/IP dilakukan pada mikrokontroller *publisher, broker* dan *subscriber* agar dapat melakukan komunikasi secara nirkabel. Konfigurasi diletakkan pada *sketch* tiap mikrokontroler. *Publisher* akan mengirimkan data sesuai data yang diterima dari *joystick* kemudian meneruskan ke *broker, broker* melanjutkan ke *subscriber* untuk dieksekusi dalam bentuk aksi pergerkan robot.

Mikrokontroler yang digunakan pada modul *subscriber* (robot) adalah ESP32. Modul ini dipilih karena memiliki beberapa keunggulan, yaitu dapat berkomunikasi secara nirkabel, memiliki jumlah pin *general purpose input output* (GPIO) yang lebih banyak, serta sudah dibekali dengan kemampuan *dual core*. Mikrokontroler tersebut akan menerima data dari *broker* dan melakukan seleksi terhadap data tersebut. Gambar 6 menunjukkan data *sheet* dari ESP32.



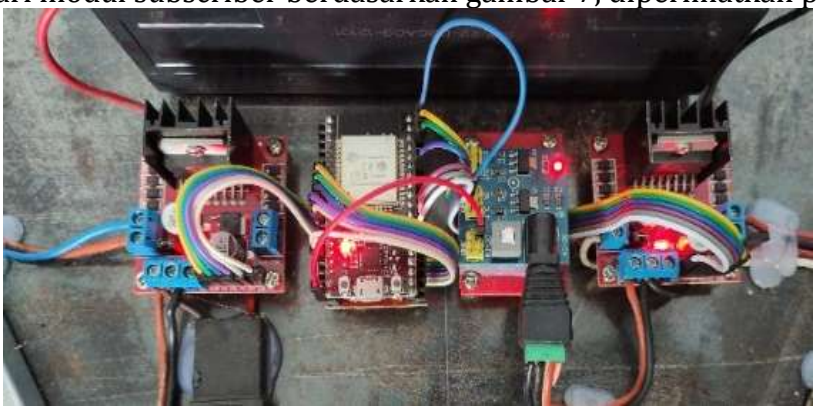
Gambar 6. Data *sheet* ESP32

Tindak lanjut dari data yang diterima dari broker adalah aksi terhadap pergerakan robot melalui motor DC dengan perantara driver motor L298N. Baterai yang digunakan pada *subscriber* berkapasitas 12V 20A untuk motor DC dan untuk mikrokontroler ESP32 diturunkan ke tegangan 5V dengan modul *stepdown* DC to DC. Skematik perangkat keras pada modul *subscriber* diperlihatkan pada gambar 7.



Gambar 7. Skematik modul *subscriber*

Implementasi dari modul subscriber berdasarkan gambar 7, diperlihatkan pada gambar 8.



Gambar 8. Modul *subscriber* (robot kameraman)

Sedangkan untuk *broker* digunakan modul *raspberry*. *Raspberry* dapat digunakan

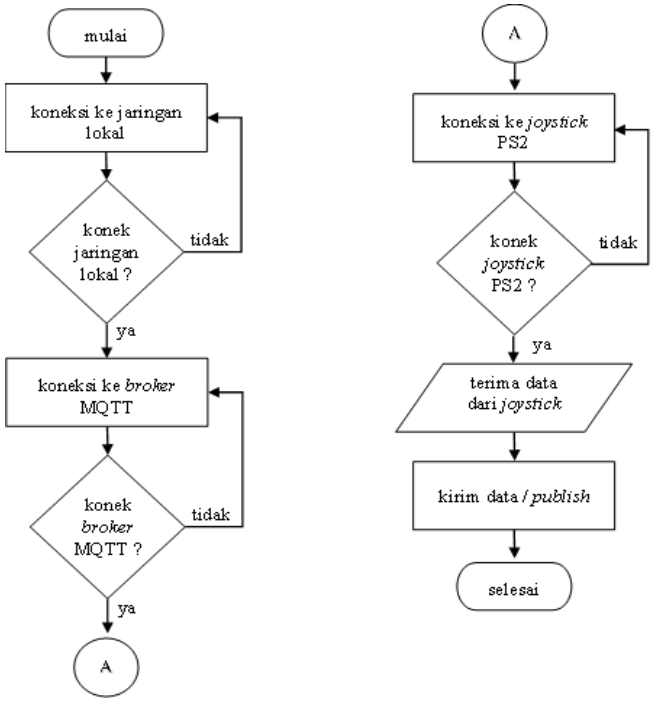
ISSN 2798-3471 (Cetak)
ISSN 2798-3641 (Online)

dalam beragam proyek, mulai dari pemrograman, pembelajaran komputer, perangkat *Internet of Things* (IoT), otomatisasi rumah, hingga server mini. Keunggulannya terletak pada fleksibilitas, kemampuan untuk mengakses kode sumbernya, dan berbagai aksesori yang tersedia untuk mendukung pengembangan proyek [8]. Pada broker digunakan perangkat *raspberry* sebagai perangkat keras dan *mosquitto* sebagai perangkat lunak. *Mosquitto* adalah salah satu perangkat lunak yang menyediakan fungsi-fungsi *broker* MQTT yang dikembangkan oleh *Eclipse Foundation*. *Mosquitto* menfasilitasi pengiriman pesan antar perangkat *publisher* dan s*ubscriber* dengan menggunakan protokol MQTT. Modul broker yang digunakan diperlihatkan pada gambar 9.

Gambar 9. Modul *broker (raspberry)*

Perancangan dan Implementasi Perangkat Lunak

Perancangan perangkat lunak menggunakan *flowchart*. *Flowchart* adalah diagram alir untuk menggambarkan alur kerja atau alur logika suatu proses. Alur pada modul *publisher* diperlihatkan gambar 10.



Gambar 10. *Flowchart* proses pada modul *publisher*

Agar dapat terhubung dengan jaringan lokal, digunakan *Internet Protocol* (IP) dan *Transport Control Protocol* (TCP) dengan koneksi nirkabel. Selain itu protokol mqtt berjalan di atas *stack* TCP/IP. Konfigurasi yang dilakukan yaitu konfigurasi statis agar IP dari *publisher*, *broker* dan *subscriber* tidak berubah-ubah pada saat perangkat di mulai ulang

(restart). Proses konfigurasi dan koneksi menggunakan protokol TCP/IP pada *sketch* sebagai berikut:

```
#include <PubSubClient.h>
//konfigurasi IP statis
IPAddress IP(192,168,1,112);  //ip publisher
IPAddress GATEWAY(192,168,1,101); //gateway
IPAddress NETMASK(255,255,255,0); //subnetmask

//koneksi ke jaringan lokal/acces point
WiFi.mode(WIFI_STA);
WiFi.config(IP, GATEWAY, NETMASK);
WiFi.begin(ssid, password);
```

Selanjutnya adalah melakukan koneksi dengan *broker* menggunakan protokol MQTT agar dapat melakukan *publish* data. Konfigurasi modul ESP8266 untuk pengiriman metode MQTT digunakan *library PubSubClient*. Konfigurasi koneksi ke *broker* dengan menentukan IP yang digunakan perangkat *broker* dan menetukan *port* yang digunakan. Port yang digunakan protokol MQTT adalah port 1883. Berikut *sketch* untuk penentuan IP *address, port* dan koneksi protokol MQTT.

```
//Menambahkan library PubSubClient
#include <PubSubClient.h>
//Konfigurasi IP add dan port server
const char* mqtt_server = "192.168.1.110"; //IP broker
#define port 1883
//koneksi ke broker
client.setServer(mqtt_server, port);
```

Sedangkan koneksi *joystick* dengan ESP8666 menggunakan protokol SPI dengan bantuan *library* PS2X_lib. Konfigurasi komunikasi *joystick* dengan ESP8266 sebagai berikut:

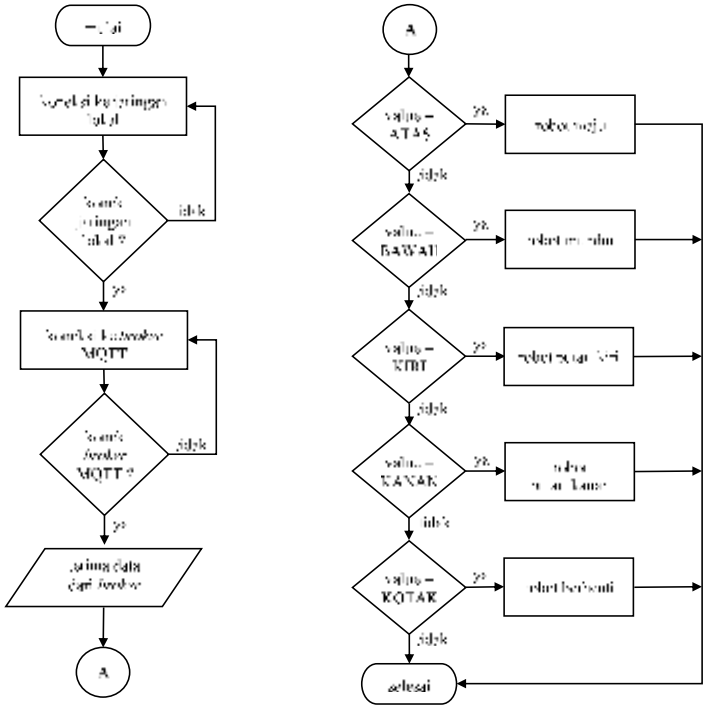
```
//Konfigurasi joystick PS2
#include <PS2X_esp_lib.h>
#define PS2_DAT 12  //D6 MISO esp8266
#define PS2_CMD 13  //D7 MOSI esp8266
#define PS2_SEL 16  //D0 SS esp8266
#define PS2_CLK 14  //D5 CLK esp8266
PS2X ps2x;
//membaca data dari joystick
ps2x.read_gamepad(true, vibrate);
```

Pada tahap akhir dilakukan proses pengiriman data dengan protokol MQTT. Proses pengiriman diawali dengan penentuan topik yang akan digunakan sebagai kata kunci berlangganan antara *publisher* dan *subcriber*. Topik yang digunakan adalah 'Tombol' diikuti dengan nilai yaitu data yang diterima dari *joystick*. Proses pengiriman data dilakukan dengan cara:

```
//proses pengiriman data
client.start();
client.publish("Tombol", value);
client.stop();
```

Selanjutnya pada modul *subscriber* atau robot, pergerakan robot terbagi atas empat gerakan yaitu, maju, mundur, putar_kiri dan putar_kanan. Robot yang digunakan menyerupai *tripot* dengan jumlah roda 3 buah. Alur proses pada modul subscriber yang digunakan sebagai berikut:

ISSN 2798-3471 (Cetak)
ISSN 2798-3641 (Online)

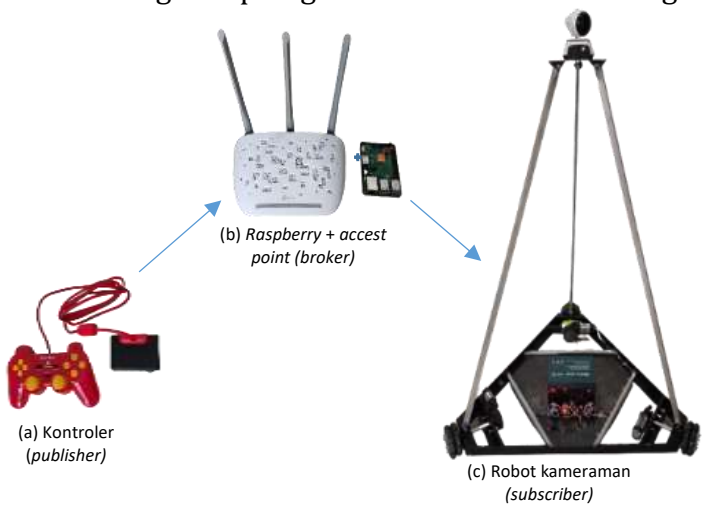


Gambar 13. *Flowchart* proses pada modul *subscriber*

Sedangkan pada modul *broker* perangkat lunak *moqquitto* dibenamkan pada *raspberry*.

Pengujian

Pengujian dilakukan dengan topologi infrastruktur sesuai dengan gambar 14.



Gambar 14. Topologi infrastruktur

Selanjutnya dilakukan kendali robot dengan memberikan perintah melalui kontroler dan mengamati aksi dari robot kameraman. Jumlah perintah yang digunakan untuk mengendalikan robot sebanyak lima perintah yaitu perintah(tombol) atas, bawah, kiri, kanan dan kotak pada *joystick*. Pergerakan pada robot sesuai dengan *flowchart* yang ditunjukkan pada gambar 13. Tombol atas untuk pergerakan robot maju, tombol bawah untuk pergerakan robot mundur, tombol kiri untuk pergerakan robot berputar ke kiri, tombol kanan untuk pergerakan robot berputar ke kanan dan tombol kotak untuk menghentikan pergerakan

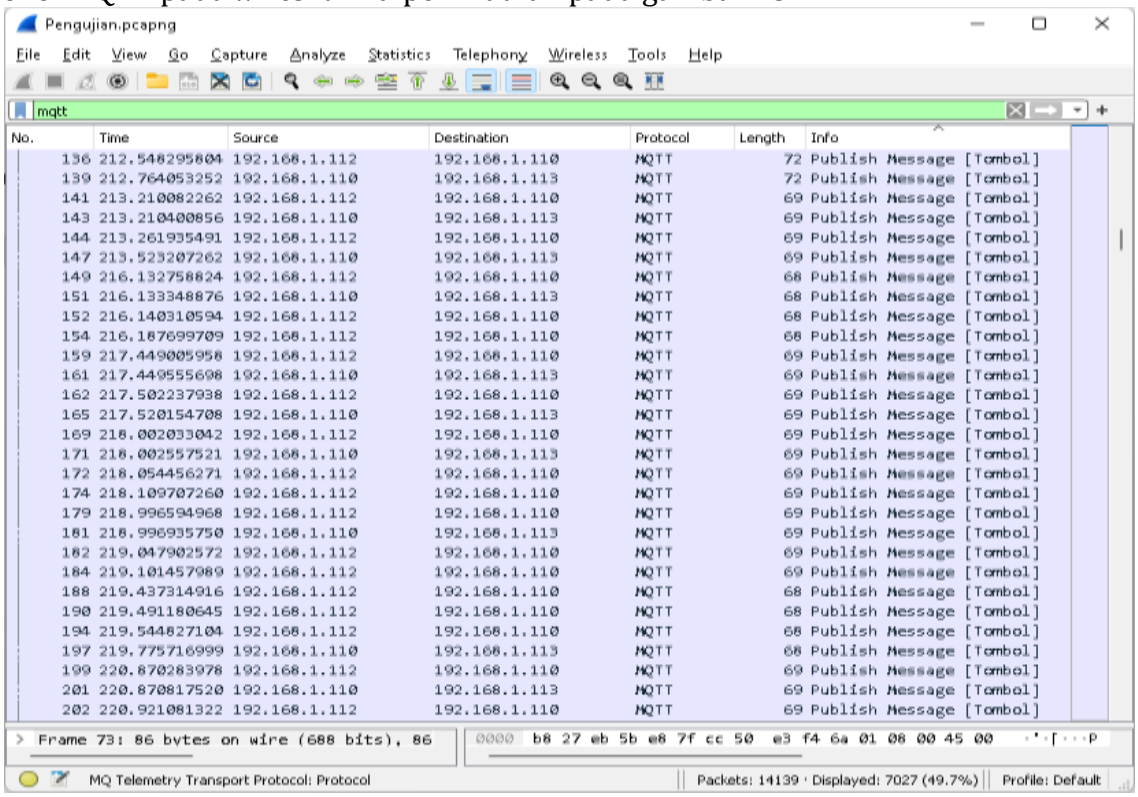
robot. Hasil pengujian kesesuaian perintah dari kontroler dengan pergerakan robot dirunjukkan pada tabel 1.

Tabel 1. Hasil pengujian akselerasi robot

Perintah	Pergerakan Robot	Kesimpulan
Atas	Maju	Sesuai
Bawah	Mundur	Sesuai
Kiri	Putar kiri	Sesuai
Kanan	Putar kanan	Sesuai
Kotak	Berhenti	Sesuai

Evaluasi

Pada tahap evaluasi dilakukan monitoring data yang dikirimkan selama proses pengujian. Alat bantu yang digunakan adalah *wireshark*. Aplikasi ini dibenamkan pada *raspberry* dan dijalankan selama proses pengujian sembari merekam lalulintas jaringan seperti alamat IP sumber, IP tujuan, protokol yang digunakan, jumlah paket yang dikirim, kapasitas paket yang dikirim, lama waktu pengiriman paket dan beberapa informasi lainya. Untuk menyeleksi paket antara kontroler dan robot, digunakan *filter* protokol pada *wireshark* dengan hanya menampilkan lalulintas jaringan dengan protokol MQTT. Hasil *filter* protokol MQTT pada *wireshark* diperlihatkan pada gambar 15.



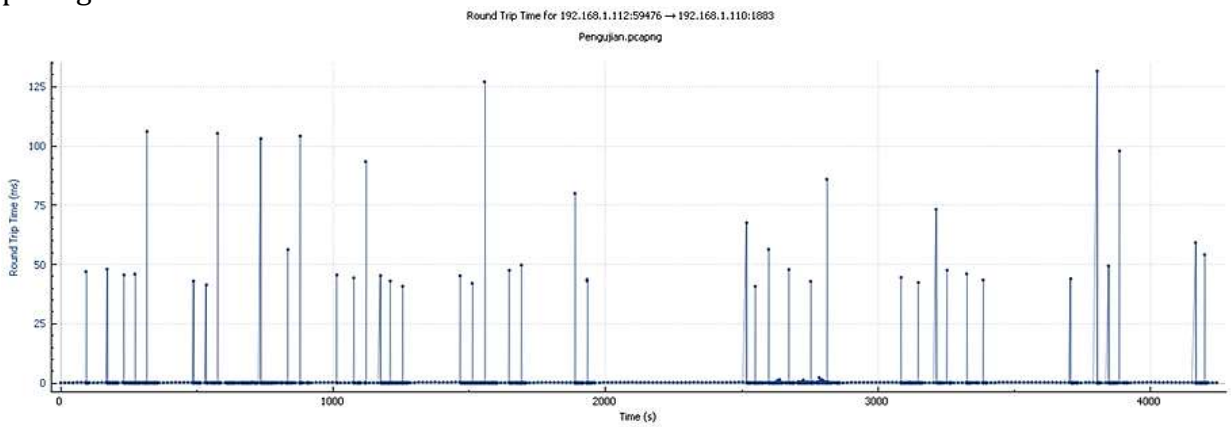
Gambar 15. Hasil *filter* protokol MQTT pada *wireshark*

Parameter yang dievaluasi adalah latensi. Latensi menunjukkan waktu yang dibutuhkan untuk mengirimkan data dari sumber ke tujuan dalam jaringan [9]. Dalam pengukuran latensi dapat menggunakan *Round-Trip Time* (RTT), yang merupakan waktu

ISSN 2798-3471 (Cetak)
ISSN 2798-3641 (Online)

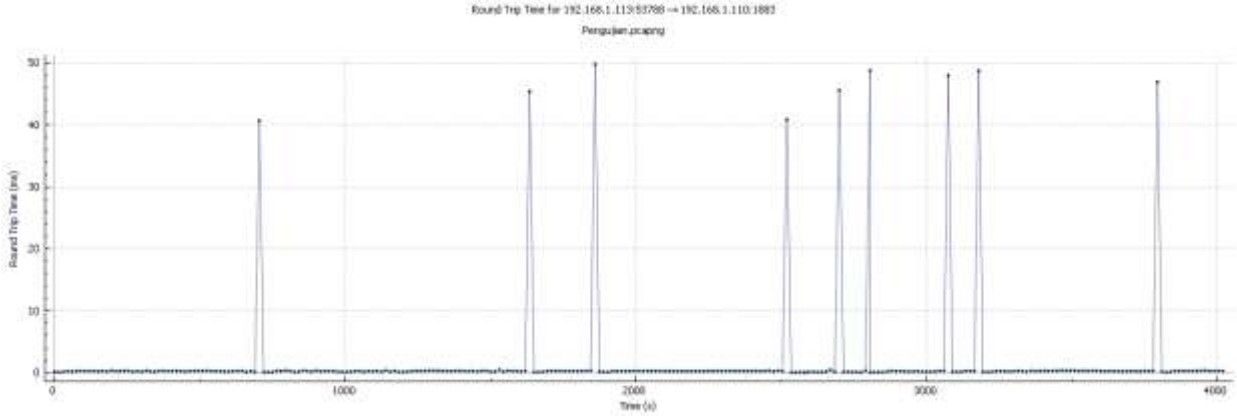
yang dibutuhkan untuk mengirimkan paket dari sumber ke tujuan dan kembali lagi ke sumber dan diukur dalam satuan milidetik [10], [11]. RTT terdiri dari dua bagian, yaitu waktu kirim *(transmit time)* dan waktu terima *(recieve time).* Waktu kirim adalah waktu saat mengirimkan paket data dari sumber ke tujuan sedangkan waktu terima adalah waktu saat menerima paket dari sumber.

Evaluasi RTT dilakukan antara kontroler *(publisher)* dengan *broker* dan robot kameraman *(subscriber)* dengan *broker*. Berdasarkan rekaman lalu lintas jaringan pada *wireshark*, diperoleh RTT antara kontroler *(publisher)* dengan *broker* seperti diperlihatkan pada gambar 16.



Gambar 16. RTT publisher dengan broker

Berdasarkan gambar 16, diperoleh nilai RTT kontroler *(publisher)* dengan broker paling cepat yaitu 0,0165 ms, paling lama 131,621 ms serta rata-rata 0,7338 ms. Sedangkan RTT untuk robot kameraman *(subscriber)* dengan broker diperlihatkan pada gambar 17.



Gambar 17. RTT subscriber dengan broker

Dari gambar 17, diperoleh nilai RTT robot kameraman *(subscriber)* dengan broker paling cepat yaitu 0,0180 ms, paling lama 49,7686 ms serta rata-rata 1,6805 ms.

KESIMPULAN

Pada penelitian ini telah berhasil diimplementasikan protokol MQTT untuk kendali robot kameraman berbasis nirkabel. Pada *publisher* digunakan mikrokontroler ESP8266 (Modul Wemos D1 Mini), pada *broker* menggunakan *raspberry* dan pada *subscriber* menggunakan mikrokontroler ESP32. Berdasarkan hasil evaluasi latensi dengan metode

round trip time (RTT) antara kontroler *(publisher)* dengan broker diperoleh rata-rata 0,7338 ms dan RTT robot kameraman *(subscriber)* dengan *broker* diperoleh rata-rata 1,68045 ms. Hal tersebut memberikan indikasi bahwa protokol MQTT dapat menjadi alternatif pengiriman data untuk mengendalikan robot secara nirkabel.

PENGAKUAN/ACKNOWLEDGEMENTS

Terima kasih kepada Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi atas dukungan finansial dan teknis dalam pelaksanaan penelitian ini. Terima kasih juga kepada Universitas Kristen Indonesia Toraja atas fasilitas dan dukungan moral bagi tim peneliti. Serta, terima kasih kepada pengelolah jurnal atas publikasi hasil penelitian dalam bentuk karya ilmiah ini.

DAFTAR PUSTAKA

[1] N. Aroon, 'Study of using MQTT cloud platform for remotely control robot and GPS tracking', in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2016, pp. 1–6.

[2] T. Yokotani and Y. Sasaki, 'Comparison with HTTP and MQTT on required network resources for IoT', in *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016, pp. 1–6.

[3] M. Z. Ahmad et al., 'Performance Analysis of Secure MQTT Communication Protocol', in *2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 2023, pp. 225–229.

[4] N. A. Sutisna and M. I. Satria, 'Development of Robotic Arm Controller Using Arduino Microcontroller and Mobile Device Application', *ROTASI*, vol. 24, no. 2, Apr. 2022, pp. 76–86.

[5] C. A. Garcia et al., 'Design of Flexible Cyber-Physical Production Systems Architecture for Industrial Robot Control', in *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, 2018, pp. 1–6.

[6] Y. Choden et al., 'Remote Controlled Rescue Robot Using ZigBee Communication', in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1–5.

[7] F. Palaha et al., 'Analisa Traffic Data Esp8266 Pada Kontrol Dan Monitoring Daya Lisrik Menggunakan Aplikasi Blynk Berbasis Arduino Nano', *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, vol. 4, no. 6, Dec. 2021, pp. 480–489.

[8] A. Jalil, 'Sistem Kendali Perangkat Elektronik Jarak Jauh Berbasis Jaringan Nirkabel Menggunakan Secure Shell (SSH) dan robot Operating System (ROS)', *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 6, Dec. 2020, pp. 1205–1212.

[9] F. Chen et al., 'Measurement and Analysis of Network Data Based on MQTT Protocol', in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, 2020, pp. 92–96.

[10] H. Idwan et al., 'Analisis Round Trip Time (RTT) Terhadap Kinerja Jaringan Wireless TCP New Reno', *Jurnal Komputer, Informasi Teknologi, dan Elektro*, vol. 3, no. 3, Aug. 2018.

[11] G. Martinez et al., 'Round Trip Time (RTT) Delay in the Internet: Analysis and Trends', *IEEE Network*, 2023, pp. 1–6.