

Peningkatan quality of experience pada permainan online multiplayer berbasis Arduino dengan menggunakan MQTT server

Enhancing the quality of experience of Arduino-based multiplayer online game using MQTT server

Gabe Dimas Wicaksana^{*)}, Maman Abdurohman, Aji Gautama Putrada

Program Studi Informatika, Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi No.1, Bandung, Indonesia 40257

Cara sitasi: G. D. Wicaksana, M. Abdurohman, and A. G. Putrada, "Peningkatan quality of experience pada permainan online multiplayer berbasis Arduino dengan menggunakan MQTT server," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 1, pp. 36-44, 2020. doi: [10.14710/jtsiskom.8.1.2020.36-44](https://doi.org/10.14710/jtsiskom.8.1.2020.36-44), [Online].

Abstract - Online multiplayer games require internet networks to play with opposing players more exciting because multiple players can fight each other. The game experiences lag, which is expressed as the quality of experience (QoE), is one of the most common problems for online multiplayer games, causing the games less exciting to play. This study examined the implementation of Message Queue Telemetry Transport (MQTT) as a communication protocol in multiplayer online games using Arduino and compared its performance against HTTP. QoE used data collected using the mean opinion score (MOS) method. The MQTT resulted in an average QoE score of 3.9 (Pingpong) and 4 (TicTacToe) MOS units, while on HTTP 3.8 (PingPong and TicTacToe). The use of the MQTT communication protocol can improve the QoE of multiplayer online game players compared to HTTP.

Keywords – MQTT; HTTP; multiplayer online game; Arduino; quality of experience

Abstrak – Permainan online multiplayer memerlukan jaringan internet agar dapat bermain lebih menarik dengan pemain lawan karena beberapa pemain bisa saling melawan satu sama lain. Salah satu kondisi permainan ini yang paling umum adalah permainan mengalami lag, yang dinyatakan sebagai quality of experience (QoE), sehingga permainan kurang menarik untuk dimainkan. Penelitian ini melakukan kajian implementasi Message Queue Telemetry Transport (MQTT) sebagai protokol komunikasi penghubung pada permainan online multiplayer di papan berbasis Arduino dan membandingkan kinerja QoE-nya terhadap HTTP. Metode mean opinion score (MOS) digunakan untuk merekam data yang diperlukan untuk menganalisis QoE. MQTT memperoleh rata-rata skor QoE sebesar 3,9

(Pingpong) dan 4 (TicTacToe) satuan MOS, sedangkan HTTP memperoleh rata-rata skor sebesar 3,8 (PingPong dan TicTacToe). Penggunaan protokol komunikasi MQTT dapat meningkatkan QoE pemain dalam permainan online karena skor rata-rata QoE-nya lebih tinggi dibandingkan dengan HTTP.

Kata kunci – MQTT; HTTP; permainan online multiplayer; Arduino; quality of experience

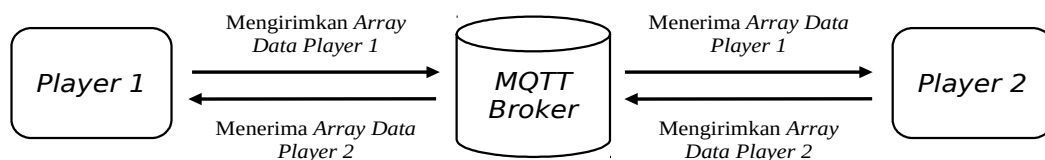
I. PENDAHULUAN

Permainan online multiplayer membutuhkan jaringan internet yang handal agar dapat bermain dengan pemain lawan lainnya [1]. Permainan ini sedang populer karena popularitas dari perangkat *handheld game console*, seperti *Nintendo Switch*. Oleh karena itu, pengembang *handheld game console* harus mengembangkan permainan yang menarik untuk dimainkan dalam kondisi apapun. Saat ini, beragam komunitas mulai mengembangkan permainan konsol 8 bit menggunakan teknologi terbaru yang menaikkan popularitas permainan ini.

Salah satu kondisi permainan online multiplayer yang paling umum adalah permainan mengalami *lag* sehingga menjadi kurang menarik untuk dimainkan, yang dinyatakan dengan *Quality of Experience (QoE)*. QoE menunjukkan tingkat kepuasan pengguna pada aplikasi atau layanan yang digunakannya [2]. Ada dua faktor penting dalam QoE yang digunakan untuk menilai permainan online multiplayer, yaitu responsif dan konsisten [3]. Suatu permainan dikatakan responsif dan konsisten secara objektif adalah dengan melihat besar *latency* yang dihasilkan. Saat ini permainan online multiplayer memiliki batas toleransi *latency* sebesar 100ms [4], [5]. Selain *latency*, jumlah variasi *latency* atau *jitter*, juga memiliki dampak yang besar pada QoE [5]. Dibandingkan dengan permainan konsol, permainan online memiliki tingkat *jitter* yang lebih besar [6].

Perangkat permainan konsol tidak bisa menjalankan permainan online multiplayer tanpa adanya sebuah penghubung antar pemain, yaitu protokol komunikasi seperti HTTP dan *Message Queue Telemetry Transport*

^{*)} Penulis korespondensi (Gabe Dimas Wicaksana)
Email: gabedimas@student.telkomuniversity.ac.id



Gambar 1. Diagram sistem dalam penelitian

(MQTT) [7]. HTTP sering digunakan sebagai protokol komunikasi utama yang sudah populer di ranah permainan online multiplayer. Harido [8] menggunakan aplikasi *Node-RED* untuk melakukan konfigurasi HTTP dengan menganalisis parameter *QoS delay*, *jitter*, *retransmission*, dan *duplicate* pada permainan multiplayer 8 bit. Skor QoE yang diperoleh berada pada rentang skor dari 3 sampai dengan 4 satuan MOS.

Di sisi lain, MQTT sebagai sebuah protokol *messaging transport* didesain untuk jaringan yang kurang *reliable* dan *bandwidth* yang rendah. MQTT bertindak sebagai *broker server*, yaitu sebagai penghubung antara pesan yang dikirim dari perangkat *publish client* kepada perangkat *subscribe client* dalam topik yang unik [9]. MQTT menunjukkan kinerja yang lebih baik dibandingkan dengan HTTP dalam aplikasi IoT dari sisi ukuran *payload*, *byte* yang ditransmisikan, dan waktu koneksi [10]. Lebih lanjut, MQTT juga mempunyai kinerja lebih baik untuk IoT daripada CoAP yang harus menjadi kestabilan transmisi [11], [12].

Kinerja dan pengaruh *lag* dalam permainan online multiplayer yang dapat menimbulkan QoE yang buruk pada pemain telah dinyatakan dalam [13]. Lag dari satu pemain tidak hanya berdampak pada pemain tersebut, namun berdampak juga ke pemain yang lain sehingga menimbulkan QoE yang buruk pada seluruh pemain. Permainan dalam satu kota yang sama memiliki nilai *Round Trip Time* (RTT) sebesar 60ms, *jitter* kurang lebih sebesar 30ms, dan *packet loss* total sebesar 1% yang membuat skor QoE menjadi antara 4 sampai dengan 5 satuan MOS. Jika ruang lingkup diperbesar menjadi skala negara, nilai RTT diperoleh sebesar 180ms, *jitter* sebesar kurang lebih 90ms, dan *packet loss* total sebesar 3%, yang sangat berdampak pada total QoE secara keseluruhan dengan skor QoE memiliki rentang antara 2 sampai dengan 4 satuan MOS.

Kinerja MQTT dalam permainan online multiplayer 8 bit dalam menentukan QoE pemain masih perlu dikaji. Penelitian ini mengkaji kinerja protokol komunikasi MQTT dan HTTP serta korelasi QoS dan QoE terhadap permainan online multiplayer 8 bit. Permainan PingPong dan TicTacToe dijalankan di atas papan berbasis Arduino. Patokan skor QoE yang digunakan berada pada rentang 3 dan 4 satuan MOS.

II. METODE PENELITIAN

A. Mean Opinion Score (MOS)

Dalam penelitian ini, penilaian QoE berdasarkan nilai MOS, yaitu berupa penilaian skalar untuk menganalisis QoE dari suatu layanan. MOS telah

banyak diterapkan di berbagai bidang layanan, seperti layanan audio, video, dan aplikasi berbasis jaringan [14]. Skor MOS diperoleh dengan melakukan perhitungan rata-rata pada nilai yang diberikan oleh responden dengan skala 1 sampai 5 sehingga dapat merepresentasikan suatu QoE dari pengguna tersebut [15]. Nilai MOS ini dihubungkan dengan nilai QoS pada protokol komunikasi yang digunakan. Parameter QoS yang dianalisis dalam protokol komunikasi MQTT ini adalah *delay*, *jitter*, *retransmission*, dan *duplicate packet*. Dengan adanya nilai QoE dan QoS, maka dapat diketahui pada batas mana parameter-parameter tersebut dapat merepresentasikan nilai QoE yang didapatkan dari responden.

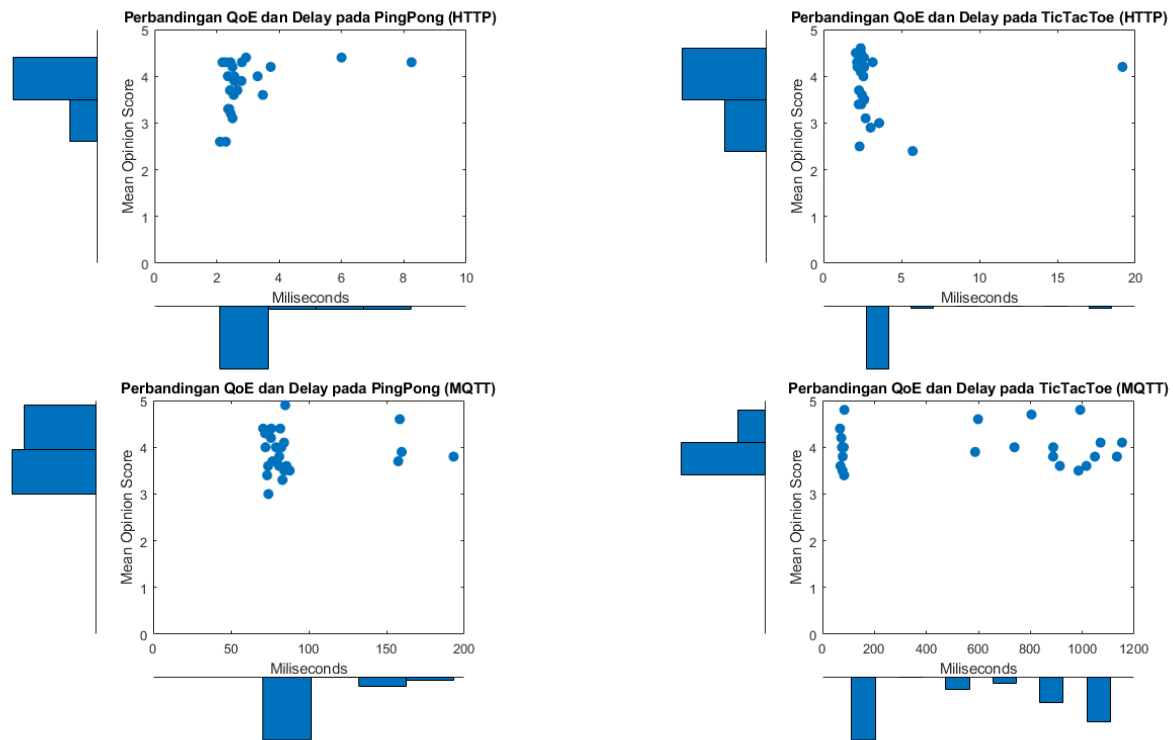
B. Desain sistem

Komponen perangkat keras yang digunakan dalam penelitian ini adalah papan NodeMCU yang berbasis Arduino. NodeMCU memiliki modul ESP8266 WiFi yang sudah terintegrasi dengan mikrokontroler sehingga sistem menjadi lebih ringkas tanpa membutuhkan modul tambahan lainnya. Komponen yang digunakan untuk menampilkan gambar dan video permainan adalah *OLED display* dengan resolusi sebesar 128 x 64 piksel. Perangkat permainan konsol dapat dioperasikan dengan menggunakan tombol berukuran 12 mm x 12 mm x 7,3 mm.

Proses transfer data yang dilakukan ditunjukkan dalam Gambar 1. Pada permainan PingPong, data yang dikirimkan adalah posisi pemain, koordinat bola, skor pemain, dan sinyal status. Sebelum melakukan pengiriman, data terlebih dahulu dijadikan sebagai *array char JSON*. Hal ini bertujuan agar performansi permainan PingPong menjadi cepat karena data yang dikirimkan hanya satu secara sekaligus daripada harus mengirimkan banyak data satu per satu.

Pada permainan TicTacToe, data yang dikirimkan adalah berupa koordinat karakter (baris dan kolom), giliran pemain, dan sinyal status. Berbeda dengan permainan PingPong, hanya beberapa data saja yang dijadikan sebagai *array char*, yaitu data koordinat karakter. Hal ini bertujuan agar pengiriman data tidak dilakukan secara terus-menerus yang dapat mengakibatkan kesalahan pada algoritme permainan.

Pada konfigurasi MQTT, tingkat QoS yang digunakan adalah QoS level 2. QoS 2 memiliki mekanisme *4-way handshake*, yaitu mekanisme untuk memastikan paket yang dikirim benar-benar sampai pada tujuan [16]. Cara kerjanya adalah *client* mengirim sebuah pesan yang ingin dikirimkan dan menunggu pesan *Acknowledgment* (PUBREC) dari *broker server*. Jika *client* tidak menerima *Acknowledgment*, maka



Gambar 2. Perbandingan QoE dengan *delay* pada PingPong dan TicTacToe (HTTP dan MQTT)

pesan dikirimkan kembali dengan *DUP flag*, yaitu jika bernilai 0 berarti pesan yang dikirim merupakan pesan yang dikirim pertama kali, sedangkan jika bernilai 1 berarti pesan yang dikirim merupakan pengiriman ulang dari pesan sebelumnya. Ketika *client* menerima *Acknowledgment*, maka *client* mengirim pesan *release message* (PUBREL).

Jika *broker server* tidak menerima *release message*, maka *broker server* mengirimkan *Acknowledgment* kembali. Jika *broker server* menerima pesan *release message*, maka pesan dari *client* dapat dikirimkan ke *broker server* dan *broker server* mengirimkan pesan *publish complete* (PUBCOMP) yang menyatakan bahwa proses pengiriman dan penerimaan pesan antar *client* dan *broker server* telah selesai. Namun, jika pesan *publish complete* tidak diterima oleh *client*, maka *client* akan mengirimkan kembali pesan *release message*. Ketika *client* menerima pesan *publish complete*, maka proses transfer pesan telah selesai dan *client* dapat menghapus pesan tersebut dari antriannya.

QoS 2 digunakan dalam penelitian ini karena tingkat *packet loss* yang rendah dan setiap pesan/paket data yang dikirimkan telah dipastikan sampai pada *broker server*. Hal ini menyebabkan pada saat permainan berlangsung pengguna dapat merasakan animasi yang konsisten sehingga meningkatkan QoE. Namun, kekurangannya adalah tingkat *delay* yang dihasilkan lebih tinggi dibandingkan dengan QoS 0 dan QoS 1.

C. Tahapan penelitian

Penelitian ini dilakukan dengan mengambil data dari 24 orang. Data QoS diambil dengan cara menangkap (*capture*) paket data dengan menggunakan aplikasi

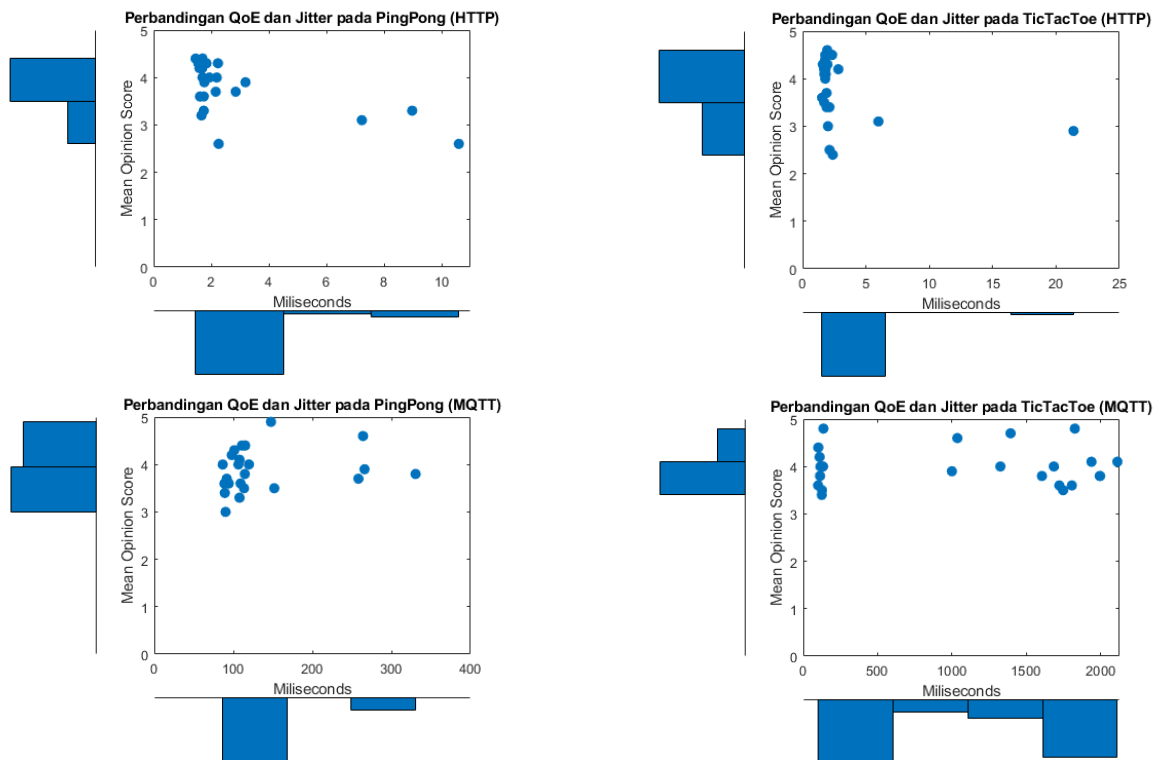
Wireshark. Data tersebut diambil sampai permainan selesai, yaitu saat salah satu pemain menang. Dari data tersebut, diambil nilai *delay*, *jitter*, *retransmission*, dan *duplicate* serta dibandingkan hasilnya dengan hasil penelitian yang telah dilakukan dengan menggunakan HTTP.

Untuk mendapatkan data QoE, setelah menyelesaikan kedua permainan (PingPong dan TicTacToe), responden diwajibkan untuk mengisi kuesioner yang berisi 10 pertanyaan untuk setiap permainan dimana setiap pertanyaan memiliki rentang skor (*rating scale*) dari satu sampai dengan lima. Setelah responden mengisi kuesioner tersebut, skor dari setiap pertanyaan tersebut dihitung dengan menggunakan rumus MOS yang sama dengan rumus rata-rata aritmatika. Dari hasil tersebut, QoE dikelompokkan menggunakan skala *5-point absolute category rating* (ACR) [17].

III. HASIL DAN PEMBAHASAN

A. Perbandingan QoE dengan delay

Perbandingan QoE dengan *delay* dalam permainan PingPong dan TicTacToe menggunakan komunikasi HTTP dan MQTT dinyatakan dalam Gambar 2. Dengan menggunakan protokol MQTT, tingkat *delay* terbesar pada permainan PingPong sebesar 193,081 ms, sedangkan tingkat *delay* terendah sebesar 70,729 ms. Rata-rata besar *delay* yang dicapai sebesar 93,671 ms. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,9 satuan MOS dan skor terendah 3 satuan MOS. Pada permainan TicTacToe, tingkat



Gambar 3. Perbandingan QoE dengan *jitter* pada PingPong dan TicTacToe (HTTP dan MQTT)

delay terbesar yaitu sebesar 1152,888 ms, sedangkan tingkat *delay* terendah sebesar 67,577 ms. Rata-rata besar *delay* yang dicapai sebesar 566,002 ms. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,8 satuan MOS dan skor terendah 3,4 satuan MOS.

Permainan PingPong memiliki skor QoE yang bervariasi, walaupun tingkat *delay* berada pada rentang yang sama. Hal ini disebabkan karena responden memiliki harapan yang tinggi sehingga mengasumsikan bahwa permainan PingPong hampir atau tidak memiliki *delay*. Ini berbeda jika dibandingkan dengan permainan *TicTacToe* yang memiliki rentang QoE lebih kecil terhadap *delay*. Hal ini disebabkan karena sifat dari permainan *TicTacToe* itu sendiri yang tidak memerlukan respons cepat sehingga pengguna yang memainkannya tidak akan merasakan *delay* tersebut. Saat pengguna diminta pendapatnya terkait permainan ini, pengguna menyatakan tidak memiliki gangguan terhadap *delay* apapun walaupun besar *delay* yang dihasilkan mencapai 1000 ms.

Dengan menggunakan protokol HTTP seperti dalam [8], tingkat *delay* terbesar pada permainan PingPong adalah sebesar 8,243 ms, sedangkan *delay* terendahnya sebesar 2,102 ms (Gambar 2). Rata-rata tingkat *delay* adalah sebesar 2,996 ms. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,4 satuan MOS, sementara skor terendah dicapai dengan skor 2,6 satuan MOS. Pada permainan *TicTacToe* menunjukkan tingkat *delay* terbesar dicapai sebesar 19,177 ms, sedangkan *delay* terendahnya dicapai sebesar 2,077 ms. Rata-rata tingkat *delay* adalah sebesar 3,333 ms. Pada

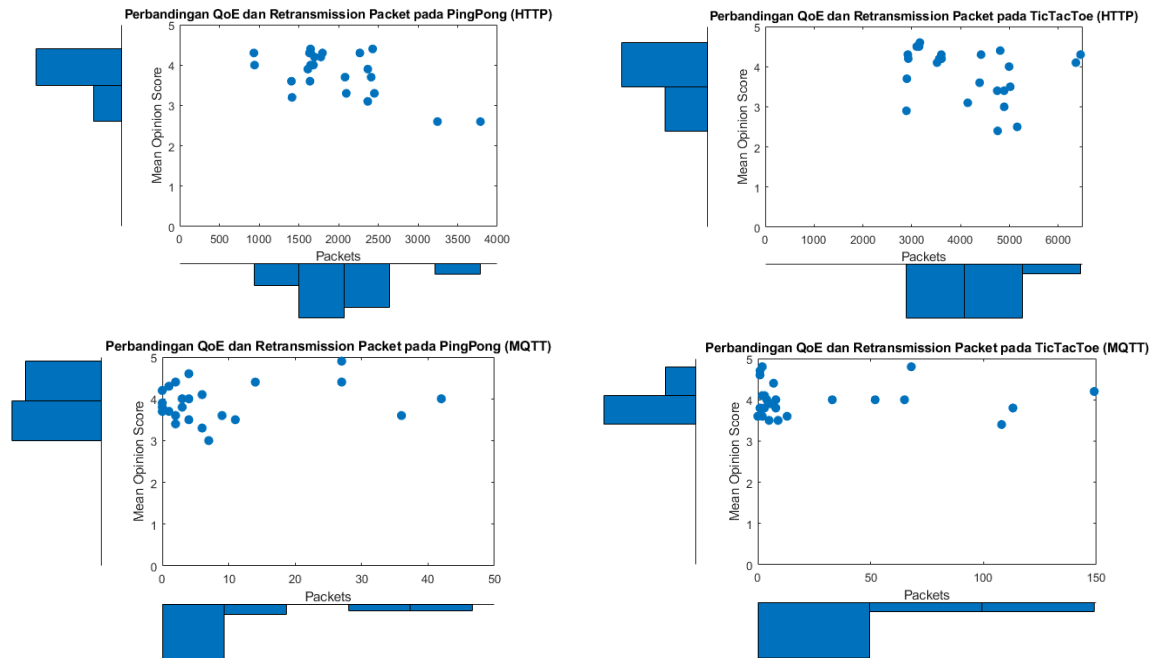
perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,6 satuan MOS dan skor terendah sebesar 2,4 satuan MOS.

Tingkat *delay* pada HTTP jauh lebih kecil dibandingkan pada MQTT. Hal ini disebabkan karena QoS yang digunakan pada MQTT adalah QoS 2 yang memprioritaskan keutuhan paket data daripada HTTP yang digunakan dalam [8]. Untuk permainan PingPong, *delay* rata-rata di bawah 100ms yang memenuhi batas toleransi [4], [5]. Walaupun demikian, skor QoE lebih besar daripada HTTP yang menunjukkan tingkat kepuasan yang lebih tinggi sesuai [2].

B. Perbandingan QoE dengan *jitter*

Perbandingan QoE dengan *jitter* dalam permainan PingPong dan *TicTacToe* menggunakan komunikasi HTTP dan MQTT dinyatakan dalam Gambar 3. Dengan menggunakan protokol MQTT, tingkat *jitter* permainan PingPong terbesar adalah sebesar 330,548 ms, sedangkan tingkat *jitter* terendah dicapai sebesar 86,658 ms. Rata-rata *jitter* yang dicapai adalah sebesar 136,570 ms. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,9 satuan MOS dan terendah 3 satuan MOS. Pada permainan *TicTacToe*, tingkat *jitter* terbesar adalah sebesar 2110,93 ms, sedangkan tingkat *jitter* terendah dicapai sebesar 100,104 ms. Rata-rata besar *jitter* yang dicapai adalah sebesar 1014,67 ms. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,8 satuan MOS dan terendah 3,4 satuan MOS.

Permainan PingPong mempunyai tingkat *jitter* yang beragam. Hal ini disebabkan karena data dikirim secara



Gambar 4. Perbandingan QoE dengan *retransmission* pada PingPong dan TicTacToe (HTTP dan MQTT)

terus menerus sehingga saat adanya *delay*, data bertubrukan dengan data yang lain dan mengakibatkan *jitter* yang beragam. Saat melakukan observasi, pemain mengeluhkan bahwa pergerakan bola pingpong terkadang patah-patah dan terkadang mulus sehingga menimbulkan skor QoE yang bervariasi antar pemain seiring pengujian berlangsung. Pada permainan TicTacToe, nilai *jitter* yang tinggi tidak akan terlalu mempengaruhi QoE pada pemain tersebut. Hal ini disebabkan karena pengiriman data pemain ke pemain yang lain hanya berlangsung ketika pemain yang satu telah selesai meletakkan karakternya dan menyelesaikan gilirannya untuk bermain.

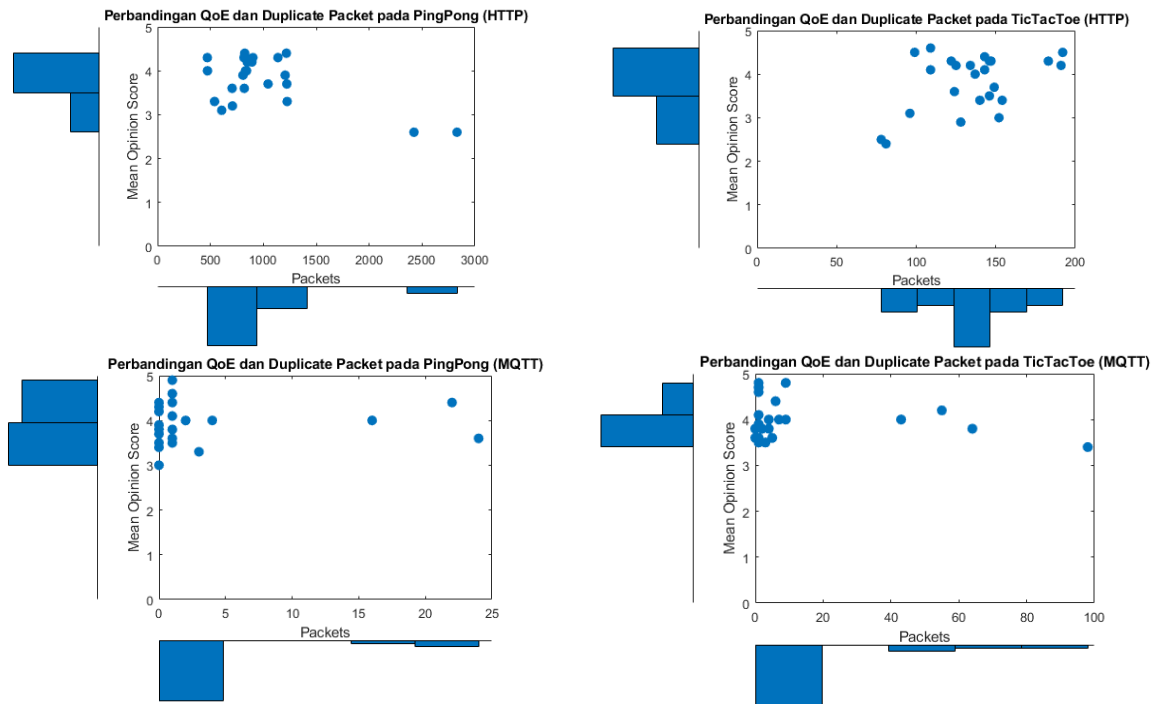
Dengan menggunakan protokol HTTP seperti dalam [8], tingkat *jitter* terbesar pada permainan PingPong adalah sebesar 10,6 ms, sedangkan *jitter* terendahnya 1,46 ms (Gambar 3). Rata-rata tingkat *jitter* adalah sebesar 2,791 ms. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,4 satuan MOS dan skor terendah 2,6 satuan MOS. Permainan TicTacToe menunjukkan tingkat *jitter* terbesar mencapai 21,377 ms, sedangkan *jitter* terendahnya 1,541 ms. Rata-rata tingkat *jitter* dalam permainan ini adalah sebesar 2,908 ms. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,6 satuan MOS dan skor terendah 2,4 satuan MOS.

Tingkat *jitter* pada kedua permainan pada HTTP jauh lebih kecil dibandingkan dengan pada MQTT. Hal ini disebabkan karena penggunaan QoS 2 pada MQTT dapat mengakibatkan tingkat *delay* yang lebih besar yang mempengaruhi *jitter* dibandingkan dengan HTTP yang digunakan dalam [8]. *Jitter* ini sangat mempengaruhi QoE keseluruhan permainan pada HTTP dan MQTT yang sesuai [5], [6].

C. Perbandingan QoE dengan *retransmission*

Perbandingan QoE dengan jumlah *retransmission packet* dalam permainan PingPong dan TicTacToe menggunakan komunikasi HTTP dan MQTT dinyatakan dalam Gambar 4. Pada permainan PingPong, jumlah *retransmission packet* terbesar adalah sebanyak 42 paket, sedangkan jumlah terkecilnya 0 paket. Rata-rata jumlah *retransmission packet* adalah sebanyak 9 paket. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,9 satuan MOS dan terendah dengan skor 3 satuan MOS. Pada permainan TicTacToe, jumlah *retransmission packet* terbesar adalah sebanyak 149 paket, sedangkan jumlah terkecilnya 0 paket data. Rata-rata jumlah *retransmission packet* adalah sebanyak 28 paket. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,8 satuan MOS dan terendah dengan skor 3,4 satuan MOS.

Permainan PingPong mempunyai nilai QoE pemain yang bagus berdasarkan jumlah *retransmission packet*. Hal ini disebabkan karena pengiriman data selalu dilakukan selama permainan sedang berlangsung untuk menghasilkan kualitas animasi yang baik. Namun, pengiriman data yang terus-menerus berdampak pada saat adanya *delay* dan *jitter*. Paket-paket data yang dikirimkan sebelumnya akan bertubrukan dengan paket data yang dikirimkan selanjutnya sehingga dapat menyebabkan *packet loss* dan TCP melakukan pengiriman ulang paket data untuk memastikan keutuhan data terjaga. Hasil pada permainan TicTacToe cukup berbeda karena jumlah *retransmission packet* data lebih banyak dibandingkan dengan permainan PingPong, yaitu dengan jumlah terbesarnya 149 paket data dibandingkan 42 paket data pada PingPong. Hal ini disebabkan karena saat sebelum permainan berlangsung, sistem permainan akan mengirimkan sinyal dari masing-



Gambar 5. Perbandingan QoE dengan *duplicate* pada PingPong dan TicTacToe (HTTP dan MQTT)

masing perangkat. Fungsi dari sinyal tersebut adalah untuk memastikan pemain bermain dalam waktu yang bersamaan. Sinyal terus dikirim dalam rentang waktu tertentu, termasuk saat permainan sudah berlangsung. Saat pemain pertama telah menyelesaikan gilirannya, paket data dari pemain pertama akan bertubrukan dengan data sinyal sebelumnya sehingga menyebabkan *retransmission* paket data.

Jika menggunakan protokol HTTP seperti dalam [8], jumlah *retransmission packet* terbesar pada permainan PingPong sebesar 3787 paket data, sedangkan jumlah terendahnya 935 paket data (Gambar 4). Rata-rata jumlah *retransmission packet* adalah sebesar 4183 paket data. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,4 satuan MOS dan skor terendah 2,6 satuan MOS. Permainan TicTacToe menunjukkan jumlah *retransmission packet* terbesar dicapai sebesar 6456 paket data, sedangkan jumlah terendahnya 2891 paket data. Rata-rata jumlah *retransmission packet* adalah sebesar 4183 paket data. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,6 satuan MOS dan skor terendah 2,4 satuan MOS.

Jumlah *retransmission packet* kedua permainan pada HTTP lebih besar dibandingkan pada MQTT. Hal ini disebabkan karena pada MQTT lebih mengutamakan keutuhan paket data dibandingkan kecepatan pada saat pengiriman paket data sehingga pada saat observasi animasi permainan pada MQTT sedikit lebih lambat dibandingkan dengan HTTP dalam [8], namun memiliki konsistensi yang sangat baik dibandingkan HTTP yang terkadang animasi yang dihasilkannya tidak mulus. Secara umum, *retransmission packet* memiliki pengaruh yang kecil terhadap QoE pemain pada MQTT, sedangkan pada HTTP pengaruhnya sangat besar.

D. Perbandingan QoE dengan *duplicate*

Perbandingan QoE dengan *duplicate* dalam permainan PingPong dan TicTacToe menggunakan komunikasi HTTP dan MQTT dinyatakan dalam Gambar 5. Pada permainan PingPong, jumlah *duplicate packet* terbesar adalah sebanyak 24 paket, sedangkan jumlah terkecilnya 0 paket. Rata-rata jumlah *duplicate packet* adalah sebanyak 3 paket. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,9 satuan MOS dan skor terendah 3 satuan MOS. Pada permainan TicTacToe, jumlah *duplicate packet* terbesar adalah sebanyak 98 paket, sedangkan jumlah terkecilnya 0 paket. Rata-rata jumlah *duplicate packet* yang terjadi sebanyak 13 paket data. Pada perbandingannya dengan skor QoE, skor tertinggi dicapai dengan skor 4,8 satuan MOS dan skor terendah 3,4 satuan MOS.

Jumlah *duplicate* paket data pada permainan PingPong dan TicTacToe menggunakan MQTT yang diperoleh saat melakukan rekap data menunjukkan jumlah yang kecil. Namun, jumlah *duplicate* pada permainan TicTacToe sedikit lebih besar dibandingkan dengan permainan PingPong. Kedua permainan menunjukkan hasil QoE yang cukup baik, yaitu pada rentang skor 3 dan 5 satuan MOS. Hal ini disebabkan karena QoS yang digunakan pada MQTT broker merupakan QoS 2, dimana paket data dikirimkan hanya sekali saja sehingga meminimalisir terjadinya *duplicate packet*. Pada pengambilan data ada beberapa *duplicate packet* yang terjadi, namun dapat diabaikan karena jumlahnya yang tidak signifikan.

Dengan menggunakan protokol HTTP seperti dalam [8], jumlah *duplicate packet* terbesar pada permainan PingPong sebesar 3787 paket data, sedangkan jumlah

terendahnya dicapai sebesar 935 paket data ([Gambar 5](#)). Rata-rata jumlah *duplicate packet* adalah sebesar 4183 paket data. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,4 satuan MOS dan skor terendah 2,6 satuan MOS. Pada permainan TicTacToe, jumlah *duplicate packet* terbesar dicapai sebesar 6456 paket data, sedangkan jumlah terendahnya 2891 paket data. Rata-rata jumlah *duplicate packet* adalah sebesar 4183 paket data. Pada perbandingannya dengan QoE, skor tertinggi dicapai dengan skor 4,6 satuan MOS dan skor terendahnya 2,4 satuan MOS.

Jumlah *duplicate packet* pada HTTP lebih besar dibandingkan dengan MQTT. Hal ini disebabkan karena pada HTTP tidak ada mekanisme yang menjamin keutuhan paket data pada saat dikirim melalui jaringan sehingga pada HTTP memungkinkan untuk memiliki jumlah *duplicate packet* yang besar. Jumlah *duplicate packet* mempunyai pengaruh yang kecil terhadap QoE pemain pada MQTT, namun mempunyai sangat berpengaruh pada HTTP. Peningkatan QoE permainan online multiplayer ini dapat diperoleh dengan menggunakan MQTT daripada HTTP dalam [\[8\]](#).

Skor QoE pada MQTT terhadap *delay*, *jitter*, *retransmission*, dan *duplicate packet* untuk setiap pengujian berada pada rentang 3 sampai 5 yang lebih tinggi daripada HTTP dalam [\[8\]](#). MQTT dapat menghadirkan permainan dengan QoE yang baik sesuai [\[2\]](#). Pengembangan dan penelitian lanjutan dapat dilakukan dengan menambahkan tipe-tipe permainan yang lain agar QoS dapat saling dibandingkan. Selain HTTP dan MQTT dalam [\[8\]](#)-[\[10\]](#), beragam protokol, seperti Rabbit-MQ, AMQP, dan CoAP dalam [\[11\]](#), [\[12\]](#), dapat diimplementasikan dan dianalisis perbandingan QoE-nya untuk permainan tersebut. Skenario pengujian berupa implementasi *delay* buatan pada MQTT juga dapat dilakukan untuk menganalisis pengaruhnya terhadap pengalaman pemain seperti dalam [\[13\]](#), [\[14\]](#). Pengujian di kondisi jaringan yang lebih bervariasi, seperti *mobile hotspot*, *public internet access*, atau jaringan internet manapun yang digunakan oleh banyak pengguna.

IV. KESIMPULAN

Penggunaan protokol komunikasi MQTT dapat meningkatkan QoE pemain permainan online multiplayer dengan rata-rata skor QoE pada MQTT lebih tinggi dibandingkan dengan HTTP. *Delay* dan *jitter* pada MQTT sangat berpengaruh pada keseluruhan QoE pemain, sedangkan *retransmission* dan *duplicate packet* data memiliki pengaruh yang kecil terhadap QoE pemain. Namun, *jitter*, *retransmission*, dan *duplicate* sangat berpengaruh terhadap QoE pemain pada HTTP.

DAFTAR PUSTAKA

[1] J. Xu, and B. W. Wah, "Consistent synchronization of action order with least noticeable delays in fast-paced multiplayer online games," *ACM Transactions on Multimedia Computing,*

Communications, and Applications, vol. 13, no. 1, pp. 1-25, 2016. doi: [10.1145/3003727](#)

[2] K. Brunnström et al., "Qualinet white paper on definitions of quality of experience," in *the Fifth Qualinet Meeting*, Novi Sad, Mar. 2013, pp. 1-25.

[3] C. Gao, H. Shen, and M. A. Babar, "Concealing jitter in multi-player online games through predictive behaviour modeling," in *IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Nanchang, China, May 2016, pp 62-67. doi: [10.1109/CSCWD.2016.7565964](#)

[4] M. Amiri, K. P. Malik, H. A. Osman, and S. Shirmohammadi, "Game-aware resource manager for home gateways," in *2016 IEEE International Symposium on Multimedia*, San Jose, USA, Dec. 2016, pp 403-404. doi: [10.1109/ISM.2016.0091](#)

[5] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "A hybrid edge-cloud architecture for reducing on-demand gaming latency," *Multimedia Systems*, vol. 20, pp. 503-519, 2014. doi: [10.1007/s00530-014-0367-z](#)

[6] U. Lampe, Q. Wu, S. Dargutev, R. Hans, A. Miede, and R. Steinmetz, "Assessing latency in cloud gaming," *Communications in Computer and Information Science*, vol. 453, pp 52-68, 2014. doi: [10.1007/978-3-319-11561-0_4](#)

[7] S. P. Jaikar and K. R. Iyer, "A survey of messaging protocols for IoT systems," *International Journal of Advanced in Management, Technology and Engineering Sciences*, vol. 8, no. 2, pp. 510-514, 2018.

[8] S. Hardo, "Peningkatan quality of experience pada multiplayer online game berbasis mikrokontroler menggunakan HTTP web server." *Bachelor thesis*, Telkom University, Indonesia, 2019.

[9] S. Lee, H. Kim, D. Hong, and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," in *2013 International Conference on Information Networking*, Bangkok, Thailand, Jan. 2013, pp 714-717. doi: [10.1109/ICOIN.2013.6496715](#)

[10] 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*, Bandung, Indonesia, Sept. 2016, pp 1-6. doi: [10.1109/ICCEREC.2016.7814989](#)

[11] D. Thangavel, X. Ma, A. Valera, H. Tan, and C. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Singapore, Singapore, Apr. 2014, pp. 1-6. doi: [10.1109/ISSNIP.2014.6827678](#)

[12] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in *2017 IEEE International Systems Engineering Symposium*, Vienna, Austria, Oct. 2017, pp. 1-7. doi: [10.1109/SysEng.2017.8088251](#)

- [13] E. Howard, C. Cooper, M. P. Wittie, S. Swinford, and Q. Yang, "Cascading impact of lag on quality of experience in cooperative multiplayer games," in *13th Annual Workshop on Network and Systems Support for Games*, Nagoya, Japan, Dec. 2014, pp. 1-6. doi: [10.1109/NetGames.2014.7008965](https://doi.org/10.1109/NetGames.2014.7008965)
- [14] T. Hoßfeld, P. Heegaard, M. Varela, S. Moller, "QoE beyond the MOS: an in-depth look at QoE via better metrics and their relation to MOS," *Quality and User Experience*, vol. 1, pp. 1-23, 2016. doi: [10.1007/s41233-016-0002-1](https://doi.org/10.1007/s41233-016-0002-1)
- [15] Mean opinion score (MOS) terminology, ITU-T Rec. P.800.1, 2016.
- [16] MQTT v5.0, OASIS standard, 2019. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/>
- [17] Mean opinion score interpretation and reporting, ITU-T Rec. P.800.2, 2013.