

# **Analisis performansi QoS MQTT pada sistem monitoring sungai**

**Tugas Akhir**  
**diajukan untuk memenuhi salah satu syarat**  
**memperoleh gelar sarjana**  
**dari Program Studi S1 Teknik Informatika**  
**Fakultas Informatika**  
**Universitas Telkom**

1103130067

Fitra Ilham



**Program Studi Sarjana S1 Teknik Informatika**  
**Fakultas Informatika**  
**Universitas Telkom**  
**Bandung**  
**2019**

**LEMBAR PENGESAHAN**

**Analisis performansi QoS MQTT pada sistem monitoring sungai**

**MQTT QoS performance analysis on river monitoring systems**

**1103130067**

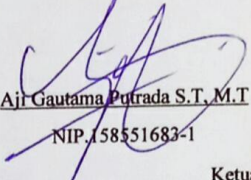
**FITRA ILHAM**

Tugas akhir ini telah diterima dan disahkan untk memenuhi sebagian syarat memperoleh gelar pada program Studi Sarjana S1 Teknik Informatika  
Fakultas Informatika  
Universitas Telkom

Bandung, 4 januari 2019

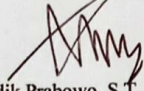
Menyetujui,

Pembimbing I,

  
Aji Gautama Putrada S.T., M.T

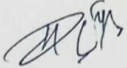
NIP.158551683-1

Pembimbing II,

  
Sidik Prabowo, S.T., M.T

NIP.15870072-1

Ketua Program Studi  
Sarjana Teknik Informatika

  
Niken Dwi Wahyu Cahayani S.T., M.Kom., Ph.D

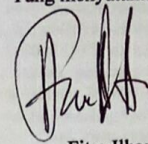
NIP.00750052

### LEMBAR PERNYATAAN

Dengan ini saya, Fitra Ilham menyatakan sesungguhnya Tugas Akhir saya dengan Judul **“Analisis performansi QoS MQTT pada sistem monitoring sungai”** beserta dengan seluruh isinya adalah merupakan hasil karya saya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya

Bandung, 4 Januari 2019

Yang menyatakan



Fitra Ilham

## Analisis performansi QoS MQTT pada sistem monitoring sungai

Fitra Ilham<sup>1</sup>, Aji Gautama Putrada<sup>2</sup>, Sidik Prabowo<sup>3</sup>

Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>fitrailham4@student.telkomuniversity.ac.id, <sup>2</sup>ajigps@telkomuniversity.ac.id,

<sup>3</sup>pakwowo@telkomuniversity.ac.id

### Abstrak

Analisis performansi QoS MQTT pada sistem monitoring sungai adalah salah satu solusi untuk menangani masalah banjir yang ada di area sungai dayeuhkolot Kab. Bandung. Oleh karena itu, dalam sistem ini, sistem ini menggunakan transmisi data yaitunya protokol MQTT dengan prinsip *wireless sensor network* dimana sensor akan mendeteksi ketinggian permukaan air kemudian sensor akan mengirimkan data ke mikrokontroler dari mikrokontroler dikirim ke *broker* MQTT kemudian, baru di *publish* ke *client* MQTT yang meng-*subscribe* topic MQTT. Pada sistem ini juga melakukan analisis QoS MQTT untuk pengiriman data dari Sensor ultrasonik ke *Raspberry Pi* yang dapat di terapkan untuk perangkat pendeteksi banjir ini agar diperoleh performansi yang optimal. Melalui pengujian ini selama 30 menit untuk setiap QoS pada satu kondisi, hasilnya waktu *delay* rata-rata pengiriman data masing-masing QoS MQTT adalah QoS0 0.943921s, QoS1 0.98576s, QoS2 1.11717s. Sedangkan *Packet Loss* rata-rata pengiriman data masing-masing QoS adalah QoS0 0.194530%, QoS1 0.24329%, QoS2 0.23604%. QoS2 sangat baik digunakan pada parameter retransmision, duplicate, dan RTT. Dikarenakan tingkat keakurasiannya di RTT lebih baik, kemudian pada duplicate dan retransmision memiliki pengiriman paket/s paling kecil dari QoS1 dan QoS2

**Kata kunci:** MQTT, QoS, Packet Loss, Delay, Throughput, Retransmision

### Abstract

MQTT QoS performance analysis on river monitoring system is one solution to deal with flooding problems in the Dayeuhkolot river area, Kab. Bandung. Therefore, in this study a case study was carried out in this final project concerning river monitoring. In designing this system, this system uses data transmission, namely the MQTT protocol with the principle of a wireless sensor network where the sensor will detect the water level then the sensor will send data to the microcontroller from the microcontroller sent to the MQTT broker then publish it to the subscribe MQTT client the topic of MQTT. This system also performs MQTT QoS analysis for sending data from ultrasonic sensors to Raspberry Pi which can be applied to this flood detection device in order to obtain optimal performance. Through this test for 30 minutes for each QoS in one condition, the result is that the average delay time of data transmission for each MQTT QoS is QoS0 0.943921s, QoS1 0.98576s, QoS2 1.11717s. While the *Packet Loss* of the average data transmission for each QoS is QoS0 0.194530%, QoS1 0.24329%, QoS2 0.23604%. QoS2 is very good for retransmission, duplicate, and RTT parameters. Because the accuracy at RTT is better, then duplicate and retransmission have the smallest packet / s deliveries of QoS1 and QoS2

**Keyword:** MQTT, QoS, Packet Loss, Delay, Throughput, Retransmision

### 1. Pendahuuan

#### Latar Belakang

Banjir adalah bencana alam yang sering terjadi diberbagai tempat di indonesia, dalam skala yang berbeda - beda dengan jumlah air yang meluap seperti, sugai citarum di daerah dayeuhkolot, Kab. Bandung. Menurut KBBI, berair banyak dan juga deras, kadang – kadang meluap [12]. Hal ini mengakibatkan volume air pada sungai, waduk air atau daerah aliran air lainnya yang melebihi kapasitas normal berakibat akumulasi air sungai yang membesar yang membuat banjir [1]. Di 2017 sebanyak 1058 rumah tergenangi oleh banjir dengan tinggi air sekitar 1,5 meter dan membuat korban banjir menjadi mengungsi. Penyebab utamanya tingginya curah hujan dan membuat sungai menjadi meluap dari kapasitasnya. Salah satu cara untuk mengatasi banjir yaitu dengan cara membuat sistem monitoring sungai yang dapat di kontrol dari jarak jauh (nirkabel) dengan menggunakan protokol MQTT untuk mengetahui tinggi air sungai, mengukur ketinggian air dengan menggunakan sensor ultasonik dan Raspberry Pi sehingga dapat memberikan informasi kepada masyarakat ketinggian air sungai. Penelitian ini mengambil referensi dari paper yang membahas banjir yang berjudul “Robust Flood Monitoring Platform using Message Queueing Telemetry Transport Protocol” pada penelitian ini membahas penanganan banjir dengan menyebar beberapa sensor di dua sungai untuk menentukan tinggi air dan deras arus air sungai, performansi MQTT pada CPU *cloud*.

MQTT (Message Queing Telemetry Transport) merupakan protokol yang mendukung konektivitas *internet of things*, MQTT menggunakan konsep *publish/subscribe* [2]. Dengan *publish/subscribe*, *subscriber* dapat menerima pesan dari *publisher* tanpa harus melakukan *request* terlebih dahulu. *Subscriber* hanya melakukan *subscribe* pada topik yang akan di *publish* oleh *publisher*.

Saat ini perkembangan teknologi berbasis *internet of things* dan sensor sudah banyak diterapkan salah satunya pada sensor ultrasonik, sensor ultrasonik berfungsi untuk mengukur ketinggian air pada permukaan air sungai, nantinya dihubungkan dengan Raspberry Pi sebagai mikrokontroler. MQTT merupakan salah satu protokol yang dapat memenuhi kebutuhan pengiriman data kepada client.

Dalam tugas akhir ini telah diimplementasikan sistem monitoring sungai yang berbasis MQTT yang terkoneksi dengan Raspberry Pi, dan sensor ultrasonik, yang mendeteksi ketinggian air sungai, yang berfokus pada QoS(Quality of Service) yang di berikan kepada pengguna layanan QoS. Hal ini diharapkan dapat digunakan untuk mengevaluasi dini sebelum terjadi banjir.

### Topik dan Batasannya

Masalah yang dibahas pada tugas akhir ini:

1. Bagaimana perancangan sistem monitoring sungai?
2. Bagaimana performansi jaringan QoS MQTT?
3. Bagaimana hasil simulasi terhadap sistem monitoring sungai secara keseluruhan?

Batasan masalah yang digunakan dalam tugas akhir ini adalah:

1. Performansi pengukuran *Throughput*, *Delay*, dan *Packet Loss*, *Duplicate*, *Retransmission*, *RTT*
2. MQTT *platform* menggunakan *platform Mosquito*.
3. MQTT-spy, HiveMQ.com dan MQTT Dashboard sebagai penampung data dan web socket client untuk melihat data pada PC dan Smartphone.
4. Wireshark sebagai *snipping* jaringan pada MQTT.
5. Penelitian dilakukan di anak sungai citarum yang berada di Batununggal, Bandung Kidul, Kota Bandung, Jawa Barat

### Tujuan

1. Mengimplemetasi sistem monitoring sungai dan berbasis *wireless sensor network* dan menganalisis performansi sistem.
2. Menganalisis performansi sistem yang telah terhubung dengan MQTT.
3. Mengetahui hasil simulasi faktor yang berpengaruh terhadap sungai.

### Organisasi Tulisan

Pada bagian 1 dijelaskan latar belakang dari masalah, rumusan masalah dan tujuan dari penelitian. Bagian 2 menjelaskan landasan teori yang terkait dengan penelitian ini. Bagian 3 menjelaskan implementasi dari metode yang digunakan. Bagian 4 menjelaskan mengenai pengujian dari dari implementasi metode tersebut. Serta bagian 5 merupakan kesimpulan.

### 2. Studi Terkait

Penelitian terhadap sistem monitoring sungai telah banyak dilakukan. Miguel A. Wister, Pablo Pancardo, Antonio J. Jara, J. A Hernandez-Nolasco, Francisco D. Acosta (2016), sistem sensor pengukuran tingkat air berdasarkan sensor ultrasonik, sensor ini mendeteksi banjir, menghitung jarak antara sensor dan air dengan menggunakan mikrokontroler menggunakan *Arduino Uno*. Data yang didapatkan dari sensor yang diproses oleh *Arduino Uno* untuk dikirimkan ke *server*. Kemudian, dari *server* akan dikirimkan ke *user* jejaring sosial seperti *Facebook* dan *twitter* bisa mengetahui tingkat air di sungai secara *real-time*.

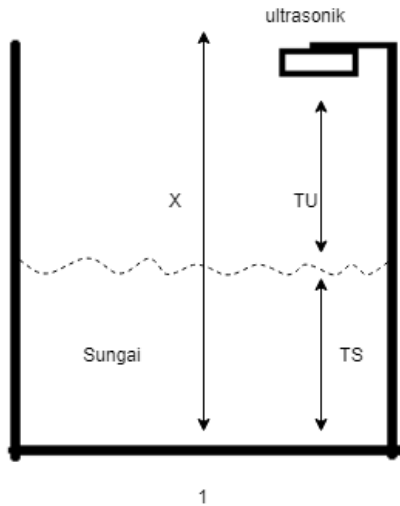
Patrice Samuel R, Aulia Arif Wardana, Albarda melakukan penelitian untuk membantu kerja BPBD (Badan Nasional Penanggulangan Bencana) untuk mengetahui ketinggian air dan arus air pada sungai untuk penanggulangan dini bencana banjir. Penelitian ini menggunakan konsep CPSS (Cyber Physical Social System), menggunakan Protokol MQTT dan *platform* pemantau banjir. Pengambilan data dari *sensing*, kemudian di simpan di *cloud* Amazon Web Service (AWS), kemudian dilihat performa keakuratan data yang diterima, menggunakan protokol MQTT. Sistem *monitoring* ini bertujuan untuk membantu BPBD dalam penanggulangan bencana banjir. Tujuan dari penelitian ini adalah performa CPU (Central Processing Unit) AWS seberapa stabil performa dari CPU komputer untuk menerima data dari sensor semakin banyak sensor yang digunakan akan membuat performa dari performa CPU menurun. Untuk mengatasi penurunan performa pada CPU.

### 3. Sistem yang dibangun

3.1 Pembuatan Alat

Melakukan pembuatan alat dengan menggunakan mikrokontroler *Raspberry Pi 2* dan Sensor ultrasonik sebagai *sensing* untuk mengambil data dari sensor tersebut dalam pengujian penanggulangan bencana dini banjir.

3.1.1 Kondisi Alat pada sungai



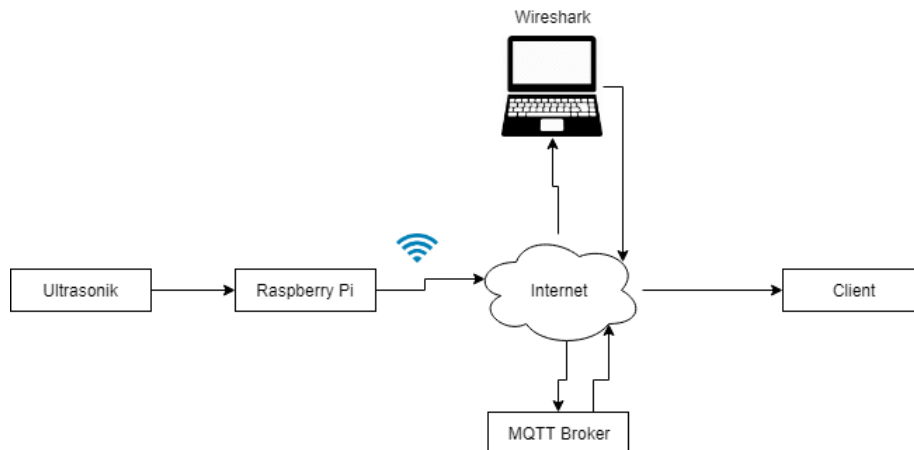
**TS = Tinggi Sungai**  
**TU = Tinggi Ultrasonik**  
**X = 263 cm (tinggi sungai dari dasar sungai ke bibir sungai)**

$$TS = X - TU$$

Gambar 1. Posisi Alat di sungai

Pada gambar 1 sensor berada dibibir sungai untuk mendapat kan tinggi sungai (TS) tinggi sungai dari dasar sungai sampai permukaan air sungai. Sehingga X dikurangi TU akan menghasilkan nilai TS.

3.1.2 Sistem Alat

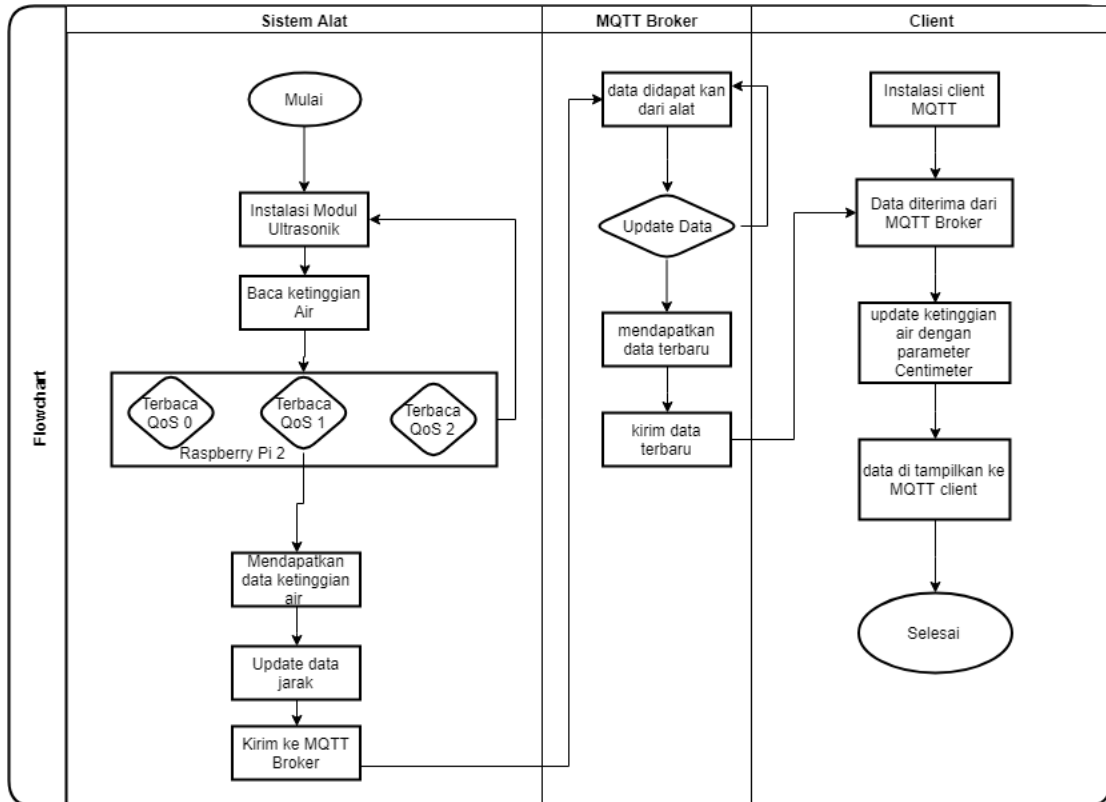


Gambar 2. Sistem Alat

Data yang di dapatkan dari ultrasonik akan dikelola oleh Raspberry Pi. Dari raspberry pi akan dikirimkan ke MQTT broker untuk di *publish* melalui internet. Data mqtt akan di analisis oleh *wireshark* oleh PC/Laptop yang sudah meng-*subscribe* terlebih dahulu topic MQTT. Sehingga *wireshark* dapat mengambil data yang ada di MQTT. Pada *client* hanya perlu meng-*subscribe* topik MQTTnya untuk mendapatkan informasi.

3.2 Rancangan Sistem

Sistem yang akan dibangun secara umum merupakan sistem yang memonitoring tinggi air sungai untuk penanggulangan bencana banjir, berdasarkan data dari sensor ultrasonik yang menggunakan protokol MQTT. Alur sistem yang akan dibuat dapat dilihat dari *flowchart* berikut:



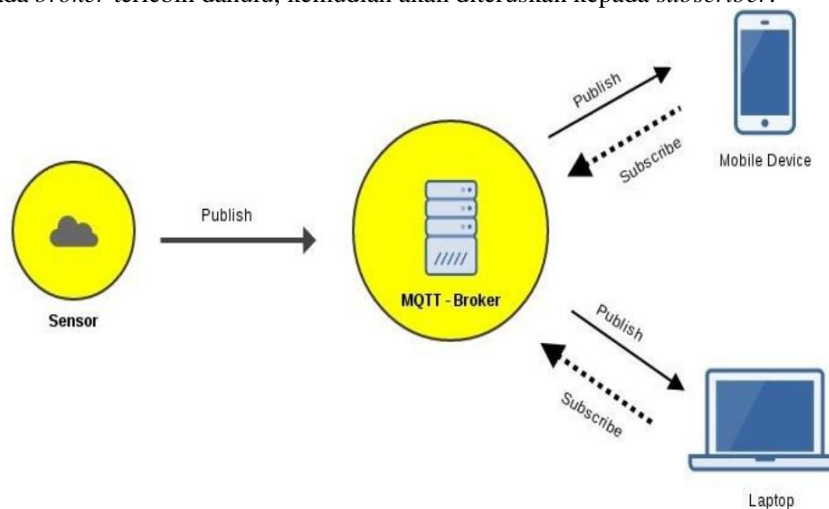
Gambar 3. Flowchart

Gambar 3 merupakan proses sistem untuk mendapatkan data yang di proses oleh sensor ultrasonik dan *Raspberry Pi 2*. Kemudian data yang di dapatkan akan akan di konfigurasi pada *Raspberry Pi 2* dalam bentuk QoS (Quality of Service) 0, 1, dan 2. Data yang telah melewati proses QoS akan dikirimkan ke *MQTT Broker* yang akan di *Publish* ke *Client* yang meng-*Subscribe* ke *MQTT Broker*. *Client* yang meng-*Subscribe* akan mendapatkan data dari *MQTT Broker*, data ketinggian banjir secara *real-time*.

3.3 Klasifikasi

3.3.1 MQTT

MQTT (Message Queue Telemetry Transport) adalah protokol yang berjalan diatas TCP/IP. MQTT menggunakan metode *publish/subscribe message*. Perangkat yang melakukan *publish message* dikenal dengan sebutan *publisher* sedangkan perangkat yang melakukan *subscribe* dikenal dengan sebutan *subscriber*. Perangkat *publisher* dan *subscriber* terhubung satu sama lain melalui penghubung yang disebut *broker*. Ketika *publisher* mengirimkan pesan, pesan tersebut akan dikirim kepada *broker* terlebih dahulu, kemudian akan diteruskan kepada *subscriber*.



Gambar 4. Ilustrasi Proses kerja MQTT

Dengan metode *publish/subscribe*, untuk mendapatkan data yang di *publish* oleh *publisher*, *subscriber* hanya perlu melakukan *subscribe* topik yang sama dengan topik yang di *publish* oleh *publisher*. Dengan demikian *subscriber* akan mendapatkan data tanpa harus melakukan *request* berulang-ulang [11]

Klasifikasi di bagi menjadi 3 untuk Quality of Service (QoS) dan perhitungan pada *Throughput*, *Packet Loss*, *Delay*.

3.3.2 QoS Level 0

Tingkat pelayanan di QoS 0 ini, tidak ada jaminan pengiriman. Penerima tidak menerima pesan *acknowledge* dan pesan tersebut tidak disimpan dan dikirimkan kembali oleh pengirim pesan. QoS0 sering disebut sebagai “fire and forget” dan memberikan jaminan yang sama dengan protokol TCP

Sender Action	Control Packet	Receiver Action
PUBLISH QoS 0, DUP=0		
	----->	
		Deliver Application Message to appropriate onward recipient(s)

Table 1. Quality of Service 0

3.3.3 QoS Level 1

Pelayanan pada QoS1 ini menjamin pengiriman pesan setidaknya satu kali kepada penerima pesan. Pengirim menyimpan pesan sampai menerima *packet* PUBACK dari *acknowledge* penerima pesan. Ini memungkinkan pesan dikirim atau pengiriman berulang-ulang.

Pengirim menggunakan *packet* indentifikasi yang membuat setiap *packet* cocok dengan *packet* PUBLISH dan *packet* PUBACK. Jika pengirim tidak menerima *packet* PUBACK di waktu yang ditetapkan, maka pengirim akan mengirim kembali *packet* PUBLISH. Ketika menerima pesan QoS1, pesan tersebut bisa langsung diproses. Misalkan jika penerima adalah *broker*, *broker* mengirim pesan ke semua *Client* yang meng-*Subscribe* *broker* dan kemudian membalas dengan *packet* PUBACK. Jika *Client* mengirim pesan PUBLISH maka akan di tetapkan sebagai pesan *duplicate* (DUP) *flag* (bendera). Pada QoS 1, bendera duplikat hanya digunakan untuk proses internal dan tidak diproses oleh *Broker* dan *Client*. Penerima pesan mengirim PUBACK, terlepas dari pengiriman *DUP flag*.

Sender Action	Control Packet	Receiver action
Store message		
Send PUBLISH QoS 1, DUP 0, <Packet Identifier>	----->	
		Initiate onward delivery of the Application Message <sup>1</sup>
	<-----	Send PUBACK <Packet Identifier>
Discard message		

Table 2. Quality of Service 1

3.3.4 QoS level 2

Layanan tertinggi MQTT yaitunya QoS level 2. Di level ini bahwasanya setiap pesan yang diterima hanya diterima satu kali oleh penerima yang tituju. QoS 2 memiliki kualitas yang paling aman dan yang paling lama. Jaminan pesan dilakukan setidaknya dua permintaan/respon dari pengirim dan penerima. Pengirim dan penerima menggunakan pengindentifikasian *packet* dari pesan PUBLISH untuk mengkoordinasikan pengiriman pesan.

Ketika penerima menerima *packet* PUBLISH QoS 2 dari pengirim, *packet* diproses apakah *packet* PUBLISH sesuai dan membalas pengirim dengan *packet* PUBREC untuk *Acknowledge* *packet* PUBLISH. Jika pengirim tidak mendapatkan *packet* PUBREC dari penerima, pengirim akan mengirim *packet* PUBLISH lagi dengan *DUP flag* sampai menerima *Acknowledge*.

Setelah pengirim menerima *packet* PUBREC dari penerima, pengirim bisa membuang *packet* PUBLISH yang awal. Pengirim menyimpan *packet* PUBREC dari penerima dan merespon dengan *packet* PUBREL. Penerima dapat membuang *packet* status tersimpan dan menjawab dengan *packet* PUBCOMP. Penerima menyelesaikan proses pengiriman *packet* PUBCOMP ke pengirim.



Penerima menyimpan referensi untuk menyimpan mengidentifikasi *packet* dari *packet* PUBLISH. Setelah pengirim menerima *packet* PUBCOMP, mengidentifikasi *packet* dari pesan yang dipublikasikan menjadi tersedia untuk digunakan kembali.

Ketika QoS 2 selesai, pengirim dan penerima memiliki konfirmasi pesan pengiriman. Jika suatu *packet* gagal di kirimkan, pengirim bertanggung jawab mengirim kembali dalam waktu yang *reasonable*. Hal ini sama dengan penfirman *packet* ke *Client* MQTT atau *Broker* MQTT, penerima harus memberikan respon setiap *packet* yang diterima pada waktu tunggu.

Sender Action	Control Packet	Receiver Action
Store message		
PUBLISH QoS 2, DUP 0 <Packet Identifier>		
	----->	
		Method A, Store message or Method B, Store <Packet Identifier> then Initiate onward delivery of the Application Message <sup>1</sup>
		PUBREC <Packet Identifier>
	<-----	
Discard message, Store PUBREC received <Packet Identifier>		
PUBREL <Packet Identifier>		
	----->	
		Method A, Initiate onward delivery of the Application Message <sup>1</sup> then discard message or Method B, Discard <Packet Identifier>
		Send PUBCOMP <Packet Identifier>
	<-----	
Discard stored state		

Table 3. Quality of Service

### 3.3.5 Sensor Ultrasonik HC - SR04

Sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

### 3.3.6 Raspberry Pi 2

*Raspberry Pi 2* adalah sebuah mini CPU yang biasa disebut mikrokontroler. *Raspberry Pi 2* memiliki prosesor, RAM, dan *port hardware* khusus. *Raspberry Pi 2* dibangun dengan menggunakan sistem operasi khusus yakni sistem operasi *Raspbian* yang merupakan bagian dari LINUX, namun didalam *Raspberry Pi 2* tetap dapat di-install sistem operasi lain seperti Windows, OSMC, Pirate OS, dsb

### 3.3.7 Wireshark

*Wireshark* adalah *tool Network Analyzer* yang digunakan oleh *Network Administrator* untuk melakukan Analisa kinerja jaringan dan melakukan kontrol lalu lintas data di jaringan. *Wireshark* dapat menangkap *packet* data yang lewat didalam jaringan untuk di analisis (*packet* yang ditangkap berupa informasi)

### 3.4 Skenario Pengujian

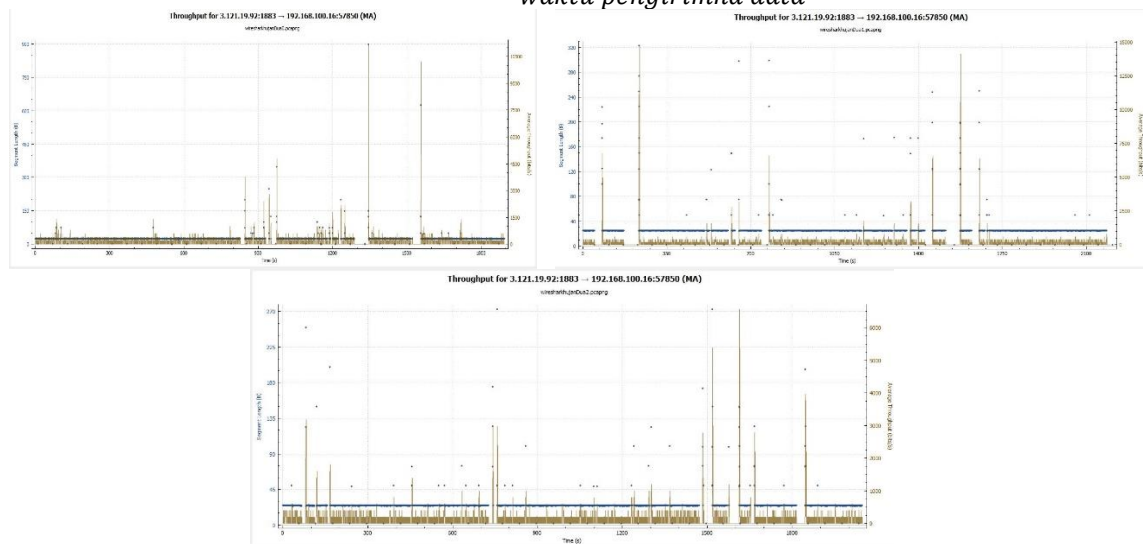
Pengujian ini bertujuan untuk mengetahui performansi sistem dengan melihat 3 jenis parameter pengujian. Pengujian ini dilakukan untuk 3 jenis level QoS MQTT yaitunya QoS0, QoS1, dan QoS2. Parameter *packet loss* diketahui dengan melakukan perhitungan rumus pada sub bab (3.3.5). dan kemudian untuk *Throughput* juga dapat didapatkan secara otomatis dengan menggunakan aplikasi wireshark untuk mengetahui perbandingan bandwidth aktual yang ditangani sistem, waktu penanganan *packet* dengan panjang segmen (Byte) tertentu. Pengujian 3 jenis parameter untuk mengetahui performansi sistem masing – masing QoS MQTT dilakuka selama masing – masing pada kondisi pagi, siang, malam, dan hujan. Lama waktu yang diambil untuk setiap pengujian QoS yaitunya 30 menit, pengujian di lakukan di atas bibir sugai yang terletak di komplek perumahan batunuggal yang mengarah ke dayeuhkolot, Kab. Bandung.

## 4. Evaluasi Hasil Pengujian

### 4.1 Throughput

*Throughput* adalah kecepatan rata-rata data yang diterima oleh suatu node dalam selang waktu pengamatan tertentu. *Throughput* merupakan *bandwidth* aktual saat itu juga dimana kita sedang melakukan koneksi. Satuan yang dimilikinya sama dengan *bandwidth* yaitu bps. Rumus untuk menghitung *Throughput*:

$$\text{Throughput} = \frac{\text{jumlah data yang dikirm}}{\text{waktu pengirimna data}} \quad (1)$$



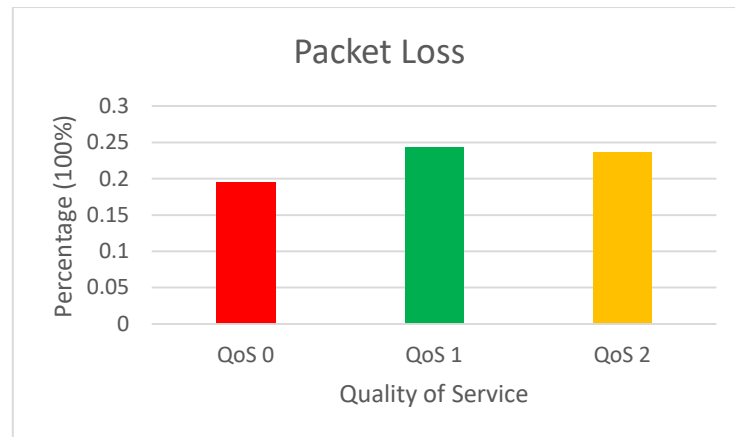
Gambar 5. Throughput

Berdasarkan gambar 4 dapat ditarik kesimpulan bah wa *throughput* yang dihasilkan oleh QoS2 lebih baik jika dibandingkan dengan *throughput* yang dihasilkan dalam penggunaan QoS0 dan QoS1 dengan melihat performansi sistem berdasarkan pengaruh waktu penanganan sistem di mana rata-rata *throughput* QoS dalam pengujian MQTT pada sistem monitoring sungai adalah QoS2 sebesar 588.167 bits/s, untuk QoS1 sebesar 688.333 bits/s, QoS0 sebesar 732.667 bits/s. Hal ini menggambarkan bahwa kecepatan pengiriman data sebenarnya dari proses pengiriman sampai diterima oleh penerima yang dapat ditangani oleh sistem dipengaruhi oleh level QoS MQTT.

### 4.2 Packet loss

*Packet loss* merupakan jumlah data yang hilang pada lalu lintas jaringan diakibatkan oleh tabrakan (collision), kapasitas jaringan, dan penurunan *packet* dikarenakan kehabisan waktu tunggu *packet*. Dalam menentukan *packet loss* dengan aplikasi *wireshark* mangunakan filter “*tcp.analysis.lost\_segment&&mqtt*” untk menentukan jumlah *packet* yang hilang dengan protokol MQTT.

$$\text{Packet Loss} = \left( \frac{\text{data yang dikirm} - \text{data yang diterima}}{\text{data yang dikirm}} \right) \times 100\% \quad (2)$$



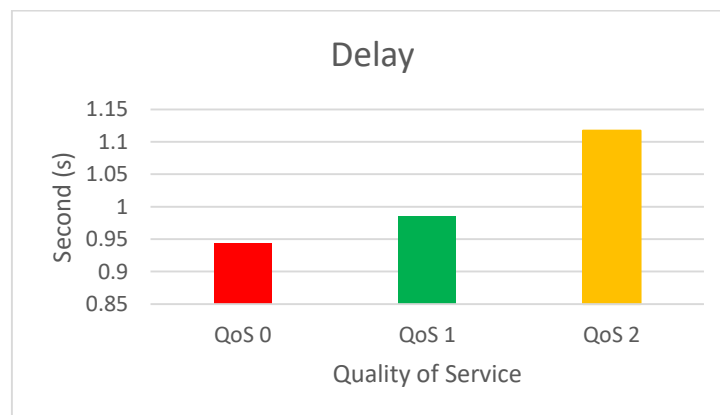
**Gambar 6.** Packet Loss

Berdasarkan gambar 5 dapat ditarik kesimpulan bahwa *packet loss* terkecil yang dialami sistem monitoring sungai dengan menerapkan MQTT sebagai media komunikasi datanya dialami dengan menerapkan QoS0 dengan *delay loss* sebesar 0.194530 %, QoS1 *delay loss* yang terjadi adalah 0.24329%, dan QoS2 adalah 0.23604%. *packet loss* dapat terjadi dikarenakan kepadatan dalam jaringan, dari hasil implementasi tersebut *packet* yang berada di jaringan saat penerapan QoS0 adalah jumlah *packet* terkecil yang ditangani sistem selama implementasi, dari gambar diatas dapat dilihat QoS 0 terkecil dari *packet loss* QoS1 dan QoS2.

#### 4.3 Delay

Menentukan waktu yang dibutuhkan *packet* yang dikirim oleh pengirim dan sampai di penerima dan dikirimkan kembali *packet* baru oleh si pengirim.

$$Delay = \frac{Time\ Span}{Packet} \quad (3)$$

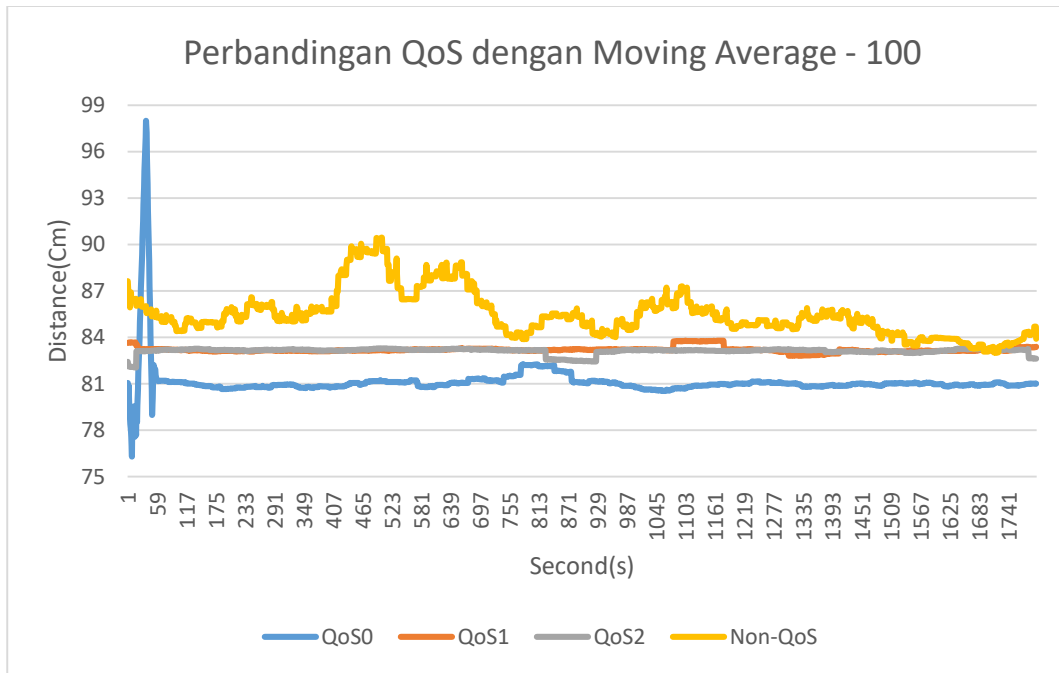


**Gambar 7.** Time Span/Packet

Pengujian *Delay* protokol MQTT untuk masing-masing QoS yang terkecil terjadi pada saat penggunaan QoS0 yakni dengan rata-rata waktu *Delay* sebesar 0.943921 *Second* (s), dengan banyak 2344 *packet*, sementara untuk QoS1 rata-rata waktu *Delay* sebesar 0.98576s, banyak *packet* yang diproses 2032 *packet*. Kemudian untuk QoS 2 rata-rata waktu *Delay* sebesar 1.11717s, dengan banyak *packet* yang diproses 2185 *packet*. Dalam penelitian ini, rata-rata *delay* QoS MQTT dengan penggunaan komunikasi wireless antara komponen menggunakan jaringan LAN akan semakin vesar siring dengan meningkatnya jumlah client yang terhubung. Jumlah *packet* yang semakin banyak belum pasti akan membesarkan *Delay* yang terjadi karena sistem dipengaruhi oleh penggunaan QoS yang memiliki masing-masing karakteristik dalam mentransmisikan *packet*-nya. Dalam penerapan protokol MQTT pada sistem monitoring sungai ini *delay* terkecil diperoleh dengan penggunaan QoS0 dikarenakan QoS0 menerapkan konsep transmisi data yang mengirimkan pesan sebanyak 1 kali tanpa melakukan pengecekan apakah pesan yang ditransmisikan telah sampai dengan sempurna di tujuannya atau tidak dan hal ini pun telah dibuktikan dalam penelitian ini. Selain itu parameter *delay* dalam pengujian ini dapat dipengaruhi oleh jarak media fisik serta waktu tunggu pada jaringan internet.

4.4 Perbandingan Data QoS dengan Non-QoS

Perbandingan *Quality of Service*(QoS) dan Non-QoS, perbandingan yang dilakukan adalah data yang didapat selama pengujian di selama 30 menit. Dengan menggunakan Moving Average atau Rata-rata bergerak adalah metode peramalan yang menghitung rata-rata suatu nilai runtut waktu dan kemudian digunakan untuk memperkirakan nilai pada periode selanjutnya. Moving Average atau Rata-rata Bergerak diperoleh melalui penjumlahan dan pencarian nilai rata-rata dari sejumlah periode tertentu, kemudian menghilangkan nilai terlamanya dan menambah nilai baru.

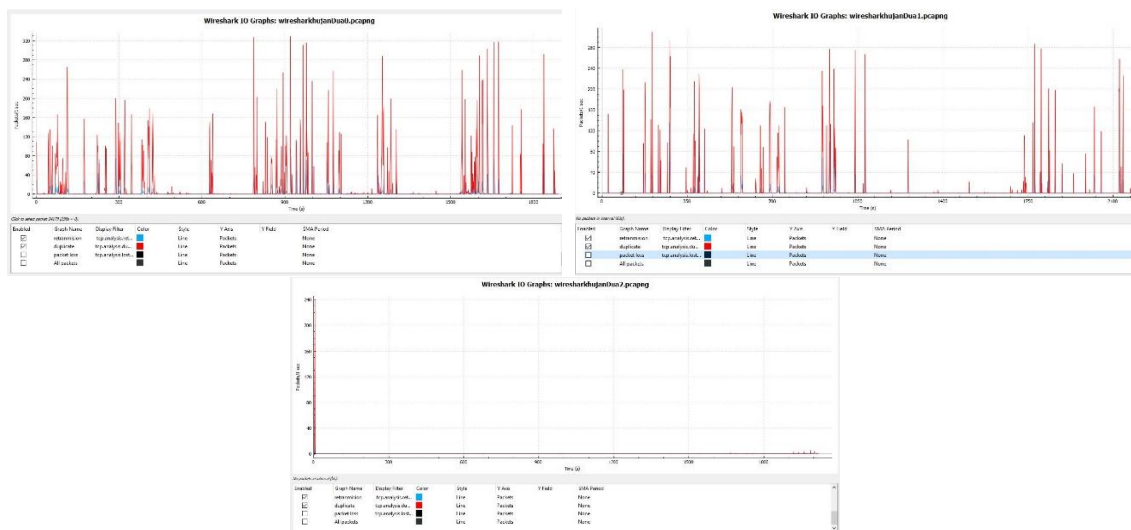


Gambar 8. Data yang diterimakan oleh setiap QoS

Dari gambar diatas (gambar 8) ini merupakan grafik data yang di terima oleh klien. Dari grafik ini bisa disimpulkan penggunaan QoS sangat di perlukan dalam monitoring sungai ini karena bisa dilihat ke akurasian data yang di terima sangat jelas bahwa Non-QoS membutuh kan skala 82 – 91 untuk nantinya terfokus ke angka 83. Dan di dibandingkan dengan QoS1 dan QoS2 dengan akurasi yang sebanding atau hampir sama. Perbedaanya hanya di QoS0.

4.5 Duplikat

Duplikasi artinya menciptakan suatu tiruan / cetakan dari aslinya. Tiruan dari duplikat tersebut harus cukup bagus untuk digunakan dalam menciptakan tiruan-tiruan selanjutnya.

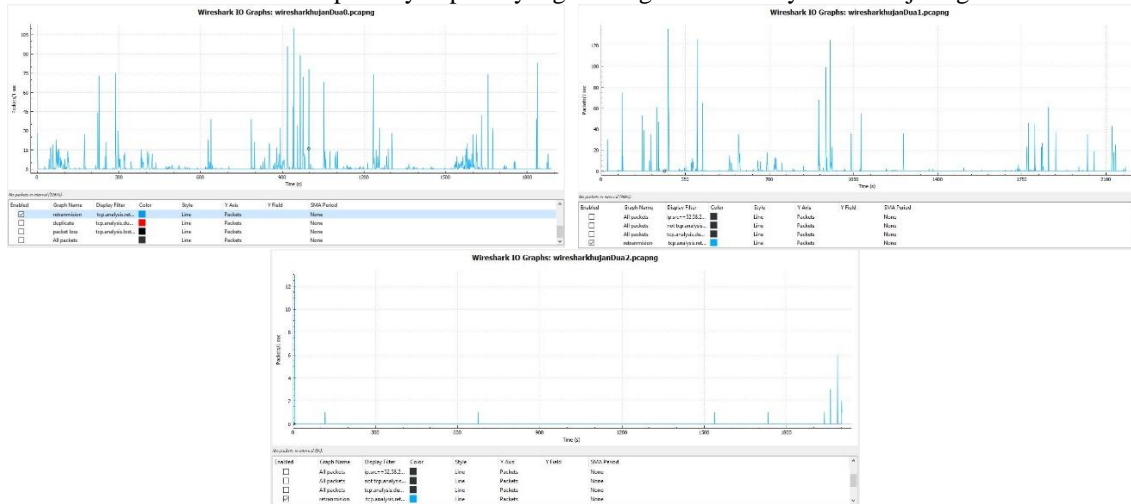


Gambar 9. Duplikat setiap Qos 0, 1, 2

Pada gambar di atas (gambar 9) hasil pengujian duplikat, grafik atas kiri adalah grafik dari hasil pengujian QoS0 dari gambar bisa dilihat banyak duplikat yang dikirimkan paling tinggi 325 paket/detik banyak dibandingkan yang lain. Gambar kanan atas QoS1, banyak duplikat paling tinggi dikirimkan 310 paket/detik, dan QoS2 di beberapa kali mengalami duplikat paling tinggi 238 paket/detik. Pada gambar bisa disimpulkan semakin tinggi dan kerapatan pada pengiriman packet/s maka semakin tinggi terjadi duplikat.

4.6 Retransmission

Retransmission adalah berupa banyak paket yang diulang karena adanya cacat di jaringan.

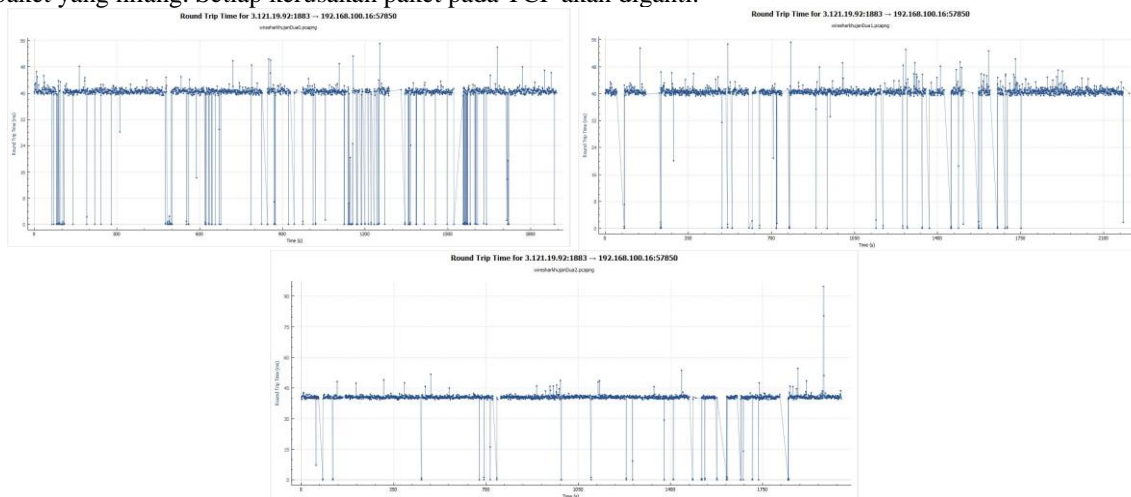


Gambar 10. Retransmission QoS 0,1, dan 2

Pada gambar di atas (gambar 10) hasil pengujian *retrasmision*, grafik atas kiri adalah grafik dari hasil pengujian QoS0 dari gambar bisa dilihat banyak *retrasmision* yang dikirimkan paling tinggi 105 paket/detik banyak dibandingkan yang lain. Gambar kanan atas QoS1, banyak duplikat paling tinggi dikirimkan 135 paket/detik, dan QoS2 di beberapa kali mengalami duplikat paling tinggi 13 paket/detik. Pada gambar bisa disimpulkan semakin tinggi dan kerapatan pada pengiriman packet/s maka semakin tinggi terjadi *retrasmision*.

4.7 RTT (Round Trip Time)

Round Trip Time adalah waktu yang dibutuhkan paket dari pengiriman sampai dengan diterimannya Ack. Setiap protocol TCP pasti memiliki RTT, karena TCP bersifat Reliable yang tidak mengijinkan adanya paket yang hilang. Setiap kerusakan paket pada TCP akan diganti.



Gambar 11. RTT setiap QoS 1, 2, 3

Pada gambar di atas (gambar 11) hasil pengujian *Round Trip Tme*, grafik atas kiri adalah grafik dari hasil pengujian QoS0 dari gambar bisa dilihat banyak *retrasmision* yang dikirimkan paling tinggi 105 paket/detik banyak dibandingkan yang lain. Gambar kanan atas QoS1, banyak duplikat paling tinggi dikirimkan 135 paket/detik, dan QoS2 di beberapa kali mengalami duplikat paling tinggi 13 paket/detik.

Pada gambar bisa disimpulkan semakin terpusatnya dan kerapatan pada pengiriman RTT maka membuat terjadi nilai RTT. Rata-rata RTT berada di 40 yang membedakannya dalam round time out pada QoS0 sering terjadi dibandingkan QoS2.

#### 4.8 Analisis Hasil Pengujian

Berdasarkan hasil pengujian Analisis performansi QoS MQTT yang diterapkan pada sistem monitoring sungai dengan melakukan implementasi uji coba selama 30 menit di setiap sesi QoS dan kategori sesi kondisi. Dari hasil pengujian QoS terbaik untuk parameter performansi Throughput, *delay*, *packet loss*, *duplicate*, *retransmission*, dan *RTT*. Delay paling kecil yaitu QoS0 dengan *delay* rata-rata 0.943921 s, hal ini dikarenakan dalam QoS0 pengiriman data dalam jaringan dilakukan tanpa melalui proses pemberitahuan data penerima apakah *packet* sudah diterima secara utuh atau belum. QoS0 menerapkan konsep *fire and forget* sehingga tidak meminta adanya pengiriman *acknowledge* hal inilah yang menyebabkan diantara 3 jenis QoS, QoS0 akan mengalami *delay* pengiriman data terkecil. *Packet loss* terkecil terjadi pada QoS0 0.194530 %. *Packet loss* terjadi diakibatkan tabrakan kapasitas jaringan, dan penurunan *packet* yang dikarenakan kehabisan waktu tunggu *packet*. Dalam penelitian ini diketahui *packet* yang berada di jaringan penerapan QoS0 adalah *packet* terkecil yang ditangani sistem selama implementasi penelitian ini dibandingkan dengan QoS1 dan QoS2 *packet loss* terkecil adalah QoS0. Untuk parameter *throughput* dari penelitian ini diketahui bahwa setiap level QoS MQTT memiliki kecepatan pengiriman data yang berbeda-beda. Dan berdasarkan pengujian *throughput* terkecil dimiliki oleh QoS2.

Parameter Duplikat terjadinya pengirim paket yang sama. retransmission, banyak paket yang di ulang karena terjadi cacat pada pengiriman paket. RTT, waktu yang dibutuhkan setiap pengiriman paket sampai ke tujuan, dan dikirimkan kembali ACK, hal yang dinilai yaitunya round time out paling sedikit. Dari duplikat, retransmission, dan RTT yang paling kecil yaitunya QoS2.

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Performansi terbaik pada analisis QoS MQTT sistem monitoring sungai komunikasi data antar perangkat berjalan dengan baik dengan QoS2 sebagai QoS terbaik setelah melakukan uji coba sistem dengan melihat beberapa parameter pengujian seperti, Duplikat data, retransmission, packet loss, Delay, Throughput, RTT.

### 5.2 Saran

- 5.2.1 Diharapkan adanya sensor yang lebih banyak agar dihasilkan tingkat akurasi yang lebih tepat.
- 5.2.2 Diharapkan adanya pembatasan data yang diproses oleh sistem agar tingkat akurasi pada QoS tidak terpengaruh terhadap data yang didapatkan yaitunya *convert base64*.
- 5.2.3 Diharapkan menggunakan QoE(Quality of Experience) untuk mengetahui kebutuhan user.
- 5.2.4 Diharapkan adanya perbandingan QoS dengan multitopik.

### Daftar Pustaka

- [1] woocara.blogspot.co.id,” pengertian banjir penyebab akibat macam - macam”[online].Available:<http://woocara.blogspot.co.id/2016/02/pengertian-banjir-penyebab-akibat-macam-macam.html>. [diakses 7 februari 2018].
- [2] Andrew banks, Rahul Gupta. mqtt version 3.1.1. Tersedia di [http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/os/mqttv3.1.1os.html#\\_Toc398718101](http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/os/mqttv3.1.1os.html#_Toc398718101) [diakses 20 september 2016].
- [3] wikipedia.org ,”Banjir”[Online].Available:<https://id.wikipedia.org/wiki/Banjir>. [diakses 7 februari 2018].
- [4] Miguel A, Pablo P, Antonio J, J A Hernandez-Nolasco, and Fransisco D, Emergency Population Warning about Floods by Social Media, 2016.
- [5] Anonim. 28 maret 2017. Sensor Ultrasonik HC-SR04. Tersedia di <http://kelas-fisika.com/2017/03/28/sensor-ultrasonik-hc-sr04/>. [diakses 4 maret 2018 ].
- [6] Mulia, Vinsen. Pengertian internet of things. Retrieved November 26, 2015, tersedia di [http://www.academia.edu/12418429/PENGERTIAN\\_INTERNET\\_OF\\_THINGS](http://www.academia.edu/12418429/PENGERTIAN_INTERNET_OF_THINGS). [diakses 7 maret 2018]
- [7] Andrew banks, Rahul Gupta. mqtt version 3.1.1. Tersedia di [http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/os/mqttv3.1.1os.html#\\_Toc398718101](http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/os/mqttv3.1.1os.html#_Toc398718101) [diakses 23 februari 2018].
- [8] Kementerian pendidikan dan kebudayaan Republik Indonesia. 2016. banjir. Tersedia di <https://kbbi.kemdikbud.go.id/entri/banjir>.
- [9] Brazell, J., Donoha, B., Dexheimer, J., Hanneman, J., & Langdon, R. (2005). M2M: The Wireless Revolution, A Techonology Forecast. *Texas State Technical College*.

- [10] ETSI. (2017, April 7). Retrieved from ETSI TS 102690: [http://www.etsi.org/deliver/etsi\\_ts/102600\\_102699/102690/02.01.01\\_60/ts\\_102690v00101p.pdf](http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v00101p.pdf). [diakses 25 februari 2018].
- [11] Anonim. Mosquitto. Tersedia di <https://mosquitto.org/>. [diakses pada 16 Februari 2018]
- [12] Anonim. 17 juni 2014. Pengetahuan Dasar dan Pemrograman Raspberry Pi. Tersedia di <https://pcontrol.wordpress.com/2014/06/17/pengetahuan-dasar-dan-pemrograman-raspberry-pi/>. [diakses 4 maret 2018].
- [13] Fiya Pyong. 9 oktober 2010. Wireshark, fungsi dan kegunaannya. Tersedia di [fiyaphyong.blogspot.com/2010/10/wireshark-fungsi-dan-kegunaannya.html](http://fiyaphyong.blogspot.com/2010/10/wireshark-fungsi-dan-kegunaannya.html). [Diakses 1 April 2018].
- [14] Team MQTT Github. 1 Juli 2018. MQTT Spy. <https://www.eclipse.org/paho/components/mqtt-spy/>. [Diakses 21 Maret 2018].
- [15] Patrice S R, Aulia A W, Albarda, Robust Flood Monitoring Platform using Message Queueing Telemetry Transport Protocol, 2017.