

PERANCANGAN INTERNET OF THINGS SISTEM MONITORING LEVEL DEBIT AIR MENGGUNAKAN MIT APP INVENTOR DAN WHATSAPP

Hamdan Sardi Selian^{1*}, Reni Rahmadewi², Yuliarman Saragih³

^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknik, Universitas Singaperbangsa, Karawang

^{1,2,3}Jln. HS Ronggo Waluyo, Kabupaten Karawang, Jawa Barat, 41361, Indonesia

email: ¹hamdan.sardi18141@student.unsika.ac.id, ²Reni.rahmadewi@staff.unsika.ac.id, ³yuliarman@staff.unsika.ac.id

Abstract — *The surrounding environment requires a technology as a means of providing information to anticipate disasters, especially flood problems. The creation of this water discharge level monitoring system is to produce a monitoring system to find out the possibility of flooding using the MIT app inventor platform which is applied in an application display in the form of water level and water discharge status with a high level of accuracy. The monitoring also uses the WhatsApp application as a notification to the public about the status of the water level. In designing to build an IoT-based water discharge level monitoring system, it is necessary to test a system that uses a test method using wireshark to see network traffic in this water discharge level monitoring system to see the Quality of Service of the WiFi delivery network to the NodeMcu as a data sender to firebase and whatsapp. The test results obtained throughput index results 0, delay 4, packet loss 4 and jitter 3. The results obtained are included in the good category according to the TIPHON standard.*

Keywords: QoS, IoT, flood, design.

Abstrak – Lingkungan sekitar membutuhkan suatu teknologi sebagai sarana pemberian informasi untuk mengantisipasi bencana khususnya permasalahan banjir. Pembuatan sistem monitoring level debit air ini untuk menghasilkan sistem monitoring supaya mengetahui kemungkinan terjadinya banjir menggunakan platform MIT app inventor yang diterapkan dalam sebuah tampilan aplikasi berupa status ketinggian air dan debit air dengan tingkat akurasi yang tinggi. Monitoring tersebut juga menggunakan aplikasi whatsapp sebagai notifikasi kepada masyarakat akan status ketinggian air tersebut. Dalam perancangan untuk membangun sistem monitoring level debit air berbasis IoT, maka dibutuhkan pengujian sistem yang menggunakan metode pengujian menggunakan wireshark untuk melihat lalu lintas jaringan yang ada di sistem monitoring level debit air ini untuk melihat Quality of Service jaringan pengiriman WiFi ke NodeMcu sebagai pengirim data ke firebase dan whatsapp. Hasil pengujian didapatkan hasil indeks Throughput 0, delay 4, packet loss 4 dan jitter 3. Hasil yang didapatkan termasuk ke kategori baik menurut standarisasi TIPHON.

Kata Kunci – QoS, IoT, Banjir, Perancangan.

I. PENDAHULUAN

Belakangan ini teknologi banyak membantu memudahkan manusia dalam mendapatkan suatu informasi apapun. Dalam hal lingkungan sekitar juga membutuhkan suatu teknologi sebagai sarana pemberian informasi untuk mengantisipasi bencana khususnya permasalahan banjir. Salah satu negara yang banyak dilanda bencana adalah Indonesia. Akibat dari luapan sungai akan menyebabkan tanah tergenang dan akan mengakibatkan banjir, sebabnya karena hujan deras atau banjir akibat kiriman dari daerah lain yang

berada di tempat yang lebih tinggi. Banjir berasal dari limpasan yang mengalir melalui sungai atau menjadi genangan. Sedangkan limpasan adalah aliran air mengalir pada permukaan tanah yang ditimbulkan oleh curah hujan setelah air mengalami infiltrasi dan evaporasi, selanjutnya mengalir menuju sungai. Sehingga limpasan mempresentasikan output dari daerah aliran sungai yang ditetapkan dengan satuan waktu [1]. Bencana banjir merupakan kejadian alam yang dapat terjadi setiap saat dan sering mengakibatkan hilangnya nyawa serta harta benda [2]. Banjir juga dapat disebabkan oleh ketinggian air yang tidak terkendali, sehingga dapat membuat reservoir meluap karena kurangnya informasi ke pengawas pintu air, ini karena pintu air saat ini sistem pengaturan [3].

Penggunaan berbagai sensor dan teknologi telah lama dibuat untuk menyaring keadaan ekologis dan bencana, seperti penggunaan perangkat pendeteksi banjir yang menggunakan Radar Doppler, namun memerlukan rencana peralatan yang rumit dan membutuhkan biaya yang besar [4]. Penelitian sebelumnya juga terdapat kerangka pengenalan banjir dengan memanfaatkan sensor ultrasonik berbasis mikrokontroler yang reaksinya masih belum cepat yaitu 5,4 detik dan selanjutnya masih menggunakan media saluran SMS [5]. Penelitian ini lebih mengarah ke Internet of Things (IoT). IoT adalah ide di mana item disematkan dengan inovasi seperti sensor dan pemrograman yang ditentukan untuk menyampaikan, mengontrol, menghubungkan, dan memperdagangkan informasi melalui berbagai gadget selama mereka terhubung dengan organisasi web [6]. Tujuan pembuatan sistem monitoring level debit air ini untuk menghasilkan sistem monitoring supaya mengetahui kemungkinan terjadinya banjir menggunakan platform MIT app inventor yang diterapkan dalam sebuah tampilan aplikasi berupa status ketinggian air dan debit air dengan tingkat akurasi yang tinggi. Monitoring tersebut juga menggunakan aplikasi whatsapp sebagai notifikasi kepada masyarakat akan status ketinggian air tersebut. Sistem monitoring banjir menggunakan arduino nano sebagai mikrokontroler dan NodeMcu Esp32 yang mengendalikan sensor ultrasonik dalam mendeteksi banjir serta ketinggiannya serta menggunakan modul komunikasi jarak jauh yaitu nrf24L01 sebagai pemancar dan penerima. Serta aplikasi MIT app inventor yang memberikan informasi mengenai statistik ketinggian air dan keadaan sensor pendeteksi air yang dimana menggunakan tiga status level air yang terprogram yaitu aman, siaga dan bahaya.

Dalam perancangan untuk membangun sistem monitoring level debit air berbasis IoT, maka dibutuhkan pengujian sistem yang menggunakan metode pengujian menggunakan wireshark untuk melihat lalu lintas jaringan yang

ada di sistem monitoring level debit air ini untuk melihat Quality of Service jaringan pengiriman WiFi ke NodeMCu sebagai pengirim data ke firebase dan whatsapp.

Penelitian ini bertujuan untuk membangun sebuah sistem monitoring level debit air. Sistem ini dapat membantu mengurangi kerugian dan dapat meminimalisir kerugian akibat banjir dan dapat dipantau secara realtime melalui aplikasi yang digunakan.

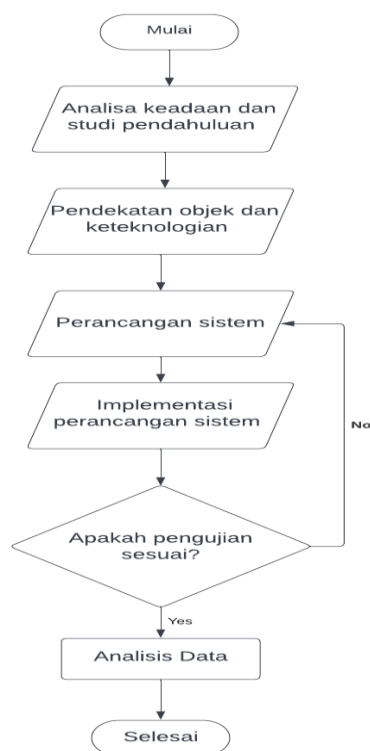
*) **penulis korespondensi:** Hamdan Sardi Selian
Email: hamdan.sardi18141@student.unsika.ac.id

II. PENELITIAN YANG TERKAIT

Penelitian yang terkait pada penelitian [7] menggunakan NodeMCU/Wemos yang telah dibangun belum mencapai tahap sempurna dan hasil pengiriman email notifikasi menggunakan IFTTT yang terhubung dengan Webhooks. Penelitian yang terkait pada penelitian [8] perangkat memberikan riwayat ketinggian air terus menerus yang diperiksa dari situs dan juga ponsel aplikasi dan selanjutnya memberikan pemberitahuan setiap ketinggian air sebagai aman atau bahaya yang realistis. Penelitian yang terkait pada penelitian [9] menggunakan wemos D1 untuk mengirimkan data ke whatsapp messenger untuk mengirimkan notifikasi adanya kebocoran LPG kepada user, hasil pengujian Quality of Service memiliki Delay rata-rata sebesar 1,052s pada jarak 40cm dan mendapatkan nilai reliability 98,43% dan nilai availability 98,46% dari skenario yang telah ditentukan.

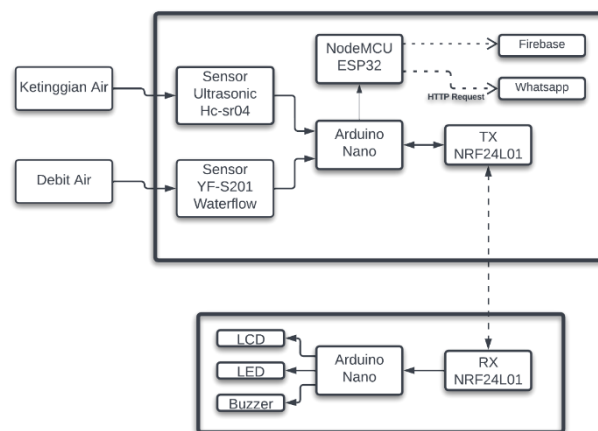
III. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini merupakan metode penelitian dengan pendekatan kuantitatif. Pendekatan kuantitatif biasanya dipakai untuk menguji suatu teori, untuk menyajikan suatu fakta atau mendeskripsikan statistik, untuk menunjukkan hubungan antar variabel, dan ada pula yang bersifat mengembangkan konsep [10]. Strategi eksplorasi ini akan mengumpulkan informasi dengan mengestimasi perangkat dan memecah sudut-sudut tertentu yang terkait dengan isu-isu yang ada sehingga dapat memberikan informasi pendukung. Tahapan yang menyertai dalam strategi eksplorasi ini harus terlihat sebagai berikut:



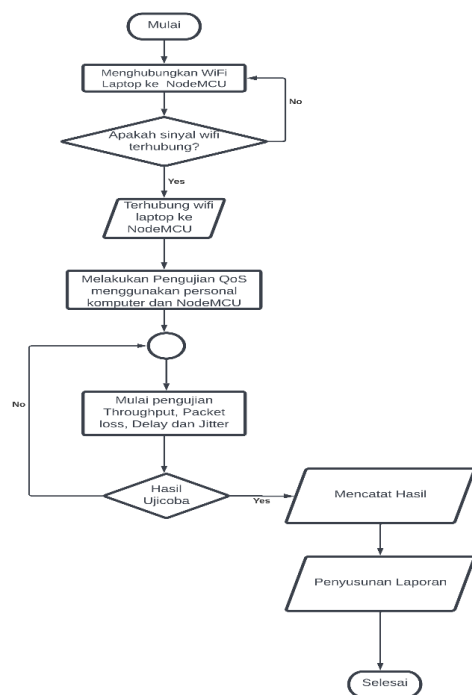
Gambar 1. Flowchart metode penelitian

A. Perancangan Desain Sistem



Gambar 2. Blok diagram perancangan sistem monitoring level debit air

Pada perancangan hardware pada blok diagram diatas merupakan perancangan untuk sistem monitoring level debit air yang didalamnya terdapat 2 blok yang berfungsi pada masing-masing bloknya untuk pengiriman dan penerima. Pada blok pengiriman terdapat arduino nano, nodemcu esp32, sensor ultrasonic hc-sr04, sensor yf-s201 waterflow dan nrf24l01 sebagai *transmitter*. Pada blok penerima terdapat arduino nano, tampilan LCD 4x20, LED, buzzer dan nrf24l01 sebagai *receiver*. Sensor yang digunakan adalah sensor waterflow ultrasonic hc-sr04 dan sensor yf-s201. Sensor ultrasonic digunakan sebagai pendeteksi ketinggian air sungai dan sensor yf-s201 waterflow digunakan untuk pendeteksi debit air sungai. Semua hasil keluaran dari sensor akan ditampilkan pada LCD yang berada di *receiver* dan hasil keluaran dari sensor juga akan dikirimkan ke firebase dan whatsapp bot. Hasil keluaran dari sensor dikirim ke firebase



Gambar 5. Flowchart pengujian QoS

Alur kerja alat ini dimulai ketika alat dinyalakan, yang pertama WiFi menghubungkan ke NodeMCU, ketika NodeMCU sudah terhubung WiFi maka bisa langsung melakukan pengujian QoS menggunakan aplikasi Wireshark untuk mencari parameter *throughput*, *packet loss*, *delay* dan *jitter*.

E. Perhitungan Throughput

Menurut panduan, nilai throughput yang telah ditentukan oleh ETSI dapat dilihat pada tabel I.

Tabel I. Standarisasi Throughput

Kategori Degradasi	Throughput	Indeks
Buruk	0 – 338 kbps	0
Kurang Baik	338 – 700 kbps	1
Cukup	700 – 1200 kbps	2
Baik	1200 – 2,1 Mbps	3
Sangat Baik	>2,1 Mbps	4

Sumber: <http://www.etsi.org>

Untuk mencari nilai Throughput harus menggunakan rumus sebagai berikut:

$$Throughput = \frac{\text{Jumlah Bytes}}{\text{Time span}}$$

F. Perhitungan Packet Loss

Menurut panduan, nilai *packet loss* yang telah ditentukan oleh ETSI dapat dilihat pada tabel 2. 3.

Tabel II. Standarisasi Packet loss

Kategori Degradasi	Packet Loss	Indeks
Buruk	>25%	1
Cukup	15 – 24%	2
Baik	3 – 14%	3
Sangat Baik	0 – 2%	4

Sumber: <http://www.etsi.org>

Untuk mencari nilai *packet loss* harus menggunakan rumus sebagai berikut:

$$Packet\ loss = \frac{(\text{Paket dikirim} - \text{paket diterima}) \times 100\%}{\text{Paket dikirim}}$$

G. Perhitungan Delay

Menurut panduan, nilai Delay yang telah ditentukan oleh ETSI dapat dilihat pada tabel III

Tabel III. Standarisasi Delay

Kategori Degradasi	Delay (milisekon)	Indeks
Buruk	>450	1
Cukup	300 – 450	2
Baik	150 – 300	3
Sangat Baik	<150	4

Sumber: <http://www.etsi.org>

Untuk mencari nilai *delay* harus menggunakan rumus sebagai berikut:

$$Delay = \text{Rata - rata Delay} = \frac{\text{Total Delay}}{\text{Total paket yang diterima}}$$

H. Perhitungan jitter

Menurut panduan, nilai Delay yang telah ditentukan oleh ETSI dapat dilihat pada tabel IV.

Tabel IV. Standarisasi Jitter

Kategori Degradasi	Jitter (milisekon)	Indeks
Buruk	125 - 225	1
Cukup	76 - 125	2
Baik	0 - 75	3
Sangat Baik	0	4

Sumber: <http://www.etsi.org>

Untuk mencari nilai *jitter* harus menggunakan rumus sebagai berikut:

$$Jitter = \frac{\text{Total Variasi Delay}}{\text{Total paket yang diterima}}$$

Untuk mencari total variasi delay adalah dengan rumus sebagai berikut:

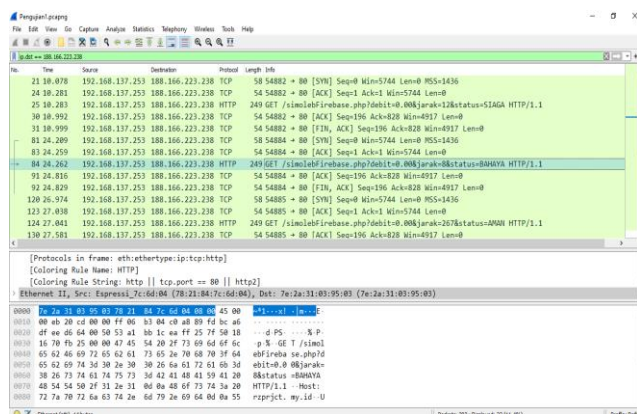
$$Jitter = Delay - (\text{Rata - rata delay})$$

IV.HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan metode Quality of Service. Metode QoS adalah metode pengujian perangkat lunak yang tujuannya untuk memeriksa lalu lintas jaringan pada sistem tanpa harus mencari secara manual. Metode ini dilakukan dengan cara menguji menggunakan Wifi yang dihubungkan ke laptop dan NodeMCU agar mendapatkan hasil yang maksimal.

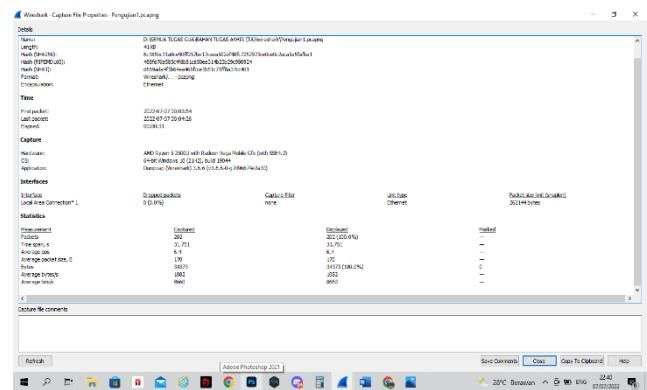
1. Pengujian *Quality of Service* pada pengujian lalu lintas jaringan pada Firebase

Pada pengujian QoS ini dilakukan untuk mengetahui kualitas koneksi jaringan yang internet yang NodeMCU V4 berikan pada aplikasi yang digunakan. Pengujian ini menggunakan bantuan dari aplikasi *wireshark* untuk mendapatkan nilai *throughput*, *packet loss*, *delay*, dan *jitter* pada sistem aplikasi SIMOLEB. Sebelum melakukan pengujian menggunakan aplikasi *wireshark*, nyalakan hotspot pada laptop, satukan jaringan NODEMCU V4 dengan jaringan hotspot Laptop. Agar kualitas jaringan terdeteksi pada aplikasi *wireshark*. Berikut merupakan gambar tampilan hasil pengujian jaringan menggunakan *wireshark*.



Gambar 6. Tampilan lalu lintas jaringan NodeMCU V4 ke firebase pada *wireshark*

Setelah mendapatkan hasil lalu lintas jaringan pada NodeMCU V4, maka nanti akan diketahui lalu lintas jaringan berupa *throughput*, *packet loss*, *delay*, dan *jitter*. Berikut merupakan *capture file properties* pada menu statistic yang ada pada aplikasi *wireshark* untuk mengetahui nilai *throughput* dan *packet loss*. Berikut merupakan hasil tampilan dari *capture file properties*:



Gambar 7. Hasil tampilan *capture file properties* pengujian firebase pada *wireshark*

Pada tampilan *capture file properties* dapat menampilkan perhitungan untuk mendapatkan *Throughput* dan *packet loss*. Pengujian ini dilakukan sebanyak 10 kali dengan jarak yang berbeda-beda. Sehingga didapatkan data sebagai berikut:

Tabel V. Hasil pengujian status pada NodeMCU V4 ke firebase

Pengujian Pengiriman			
Pengujian	NodeMCU V4	Firestore	Wireshark
1	Terkirim	Menerima	Terbaca
2	Terkirim	Menerima	Terbaca
3	Terkirim	Menerima	Terbaca
4	Terkirim	Menerima	Terbaca
5	Terkirim	Menerima	Terbaca
6	Terkirim	Menerima	Terbaca
7	Terkirim	Menerima	Terbaca
8	Terkirim	Menerima	Terbaca
9	Terkirim	Menerima	Terbaca
10	Terkirim	Menerima	Terbaca

Dari pengujian diatas yang dilakukan sebanyak 10 kali yaitu dengan pembacaan sensor jarak dan debit didapatkan lalu lintas jaringan pada NodeMCU V4 yang mengirim ke firebase terbaca di *wireshark* dengan *protocol* HTTP dan TCP.

Pengujian selanjutnya yaitu mencari *throughput*, *packet loss*, *delay*, dan *jitter* dengan metode pengujian *Quality of Service (QoS)*. Berikut merupakan hasil pengujian yang sudah dilakukan:

Tabel VI. Hasil Pengujian Pengiriman data ke firebase Dengan Jarak yang berbeda-beda

Pengujian Pengiriman Pada Jarak yang berbeda-beda				
Pengujian	Throughput (Kbps)	Packet loss	Delay (ms)	Jitter (ms)
1	8,66	0%	157,1782	15,01
2	17,7	0%	68,96648	69,82682
3	16,85	0%	73,46984	74,11429
4	17,5	0%	76,88571	77,24286
5	14,7	0%	91,42081	92,10407
6	17	0%	70,82222	71,72889
7	18,2	0%	66,1658	64,58031

8	18	0%	67,64407	67,63983
9	17,5	0%	72,26289	73,19072
10	10,24	0%	132,8207	132,7717
Total	156,35	0%	877,6367	738,2095
Rata-Rata	15,635	0%	87,76367	73,82095

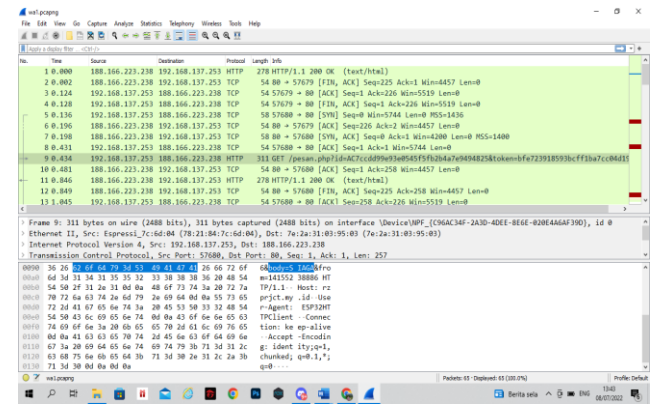
Hasil dari pengujian Quality of Service yang dilakukan sebanyak 10 kali pada setiap parameter pengujian *throughput*, *packet loss*, *delay*, dan *jitter* dengan jarak yang berbeda-beda diketahui hasil dari setiap parameter seperti tabel diatas. Pada *throughput* yang dihasilkan yaitu sebesar 15,635 Kbps. Keluaran *throughput* tersebut termasuk kekategori buruk menurut standar TIPHON. menandakan transfer data efektif pada transmitter dan receivernya baik. Pada *packet loss* data semuanya terkirim dengan baik tanpa ada yang hilang satupun, selama 10 kali pengujian semuanya dapat diterima dengan baik dan masuk kedalam jaringan firebase. Keluaran *packet loss* tersebut termasuk kekategori sangat baik menurut standar TIPHON.

Pada pengujian delay menghasilkan keluaran delay yang terbaca pada wireshark dari rata-rata delay yang sudah dilakukan pengujian 10 kali menghasilkan delay 87,76367 ms. Keluaran delay tersebut termasuk sangat baik menurut standar standar TIPHON.

Pada pengujian jitter menghasilkan keluaran jitter yang terbaca pada wireshark dari rata-rata jitter yang sudah dilakukan pengujian sebanyak 10 kali menghasilkan jitter sebesar 73,82095 ms. Keluaran jitter tersebut termasuk kekategori baik standar TIPHON.

2. Pengujian Quality of Service pada pengujian lalu lintas jaringan Whatsapp

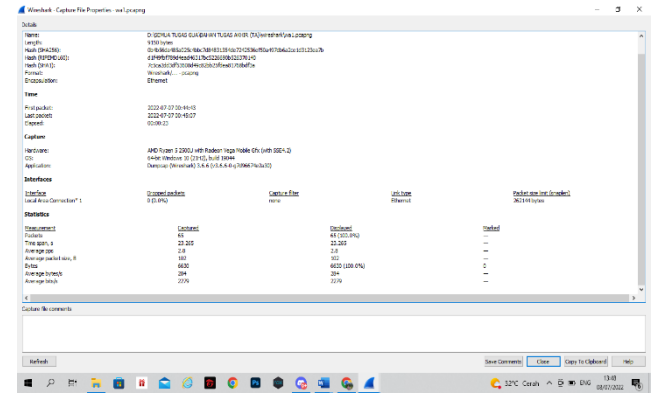
Pengujian QoS pada jaringan whatsapp dilakukan untuk mengecek apakah kualitas pengiriman data dari NodeMCU V4 baik atau tidaknya. Supaya mengantisipasi segala jenis gangguan pengiriman jaringan untuk mengirim data ke whatsapp yang digunakan pada sistem monitoring level debit air. Berikut merupakan gambar tampilan hasil pengujian jaringan menggunakan wireshark.



Gambar 8. Tampilan lalu lintas jaringan NodeMCU V4 ke whatsapp pada wireshark

Setelah mendapatkan hasil lalu lintas jaringan pada NodeMCU V4, maka nanti akan diketahui lalu lintas jaringan berupa *throughput*, *packet loss*, *delay*, dan *jitter*. Berikut merupakan *capture file properties* pada menu statistic yang ada pada aplikasi wireshark untuk mengetahui nilai

throughput dan *packet loss*. Berikut merupakan hasil tampilan dari *capture file properties*:



Gambar 9. Hasil tampilan capture file properties pengujian whatsapp pada wireshark

Pada tampilan capture file properties dapat menampilkan perhitungan untuk mendapatkan *Throughput* dan *packet loss*. Pengujian ini dilakukan sebanyak 10 kali dengan jarak yang berbeda-beda. Sehingga didapatkan data sebagai berikut:

Tabel VII. Hasil pengujian status pada NodeMCU V4 ke whatsapp

Pengujian Pengiriman			
Pengujian	NodeMCU V4	Whatsapp	Wireshark
1	Terkirim	Menerima	Terbaca
2	Terkirim	Menerima	Terbaca
3	Terkirim	Menerima	Terbaca
4	Terkirim	Menerima	Terbaca
5	Terkirim	Menerima	Terbaca
6	Terkirim	Menerima	Terbaca
7	Terkirim	Menerima	Terbaca
8	Terkirim	Menerima	Terbaca
9	Terkirim	Menerima	Terbaca
10	Terkirim	Menerima	Terbaca

Dari pengujian diatas yang dilakukan sebanyak 10 kali yaitu dengan pembacaan sensor jarak dan debit didapatkan lalu lintas jaringan pada NodeMCU V4 yang mengirim ke firebase terbaca di wireshark dengan *protocol* HTTP dan TCP.

Pengujian selanjutnya yaitu mencari *throughput*, *packet loss*, *delay*, dan *jitter* dengan metode pengujian *Quality of Service (QoS)*. Berikut merupakan hasil pengujian yang sudah dilakukan:

Tabel VIII. Hasil Pengujian Pengiriman data ke whatsapp Dengan Jarak yang berbeda-beda

Pengujian Pengiriman Pada Jarak yang berbeda-beda				
Pengujian	Throughput (Kbps)	Packet loss	Delay (ms)	Jitter (ms)
1	2,279	0%	357,9077	387,8
2	3,092	0%	268,7813	270,4531
3	1,99	0%	409,9714	251,4571
4	2,618	0%	308,2807	308,2105

5	6,844	0%	118,5455	119,1818
6	5,332	0%	153,4583	153,9896
7	4,122	0%	199,8923	200,0769
8	2,578	0%	320,7143	332,0714
9	5,346	0%	153,9266	153,6514
10	5,092	0%	162,0137	189,5616
Total	39,293	0%	2453,492	2366,453
Rata-Rata	3,9293	0%	245,3492	236,6453

Hasil dari pengujian Quality of Service yang dilakukan sebanyak 10 kali pada setiap parameter pengujian *throughput*, *packet loss*, *delay*, dan *jitter* dengan jarak yang berbeda-beda diketahui hasil dari setiap parameter seperti tabel diatas. Pada *throughput* yang dihasilkan yaitu sebesar 3,9293 Kbps menandakan transfer data efektif pada transmitter dan receivernya cukup. Keluaran *throughput* tersebut termasuk buruk menurut standar TIPHON. Pada *packet loss* data semuanya terkirim dengan baik tanpa ada yang hilang satupun, selama 10 kali pengujian semuanya dapat diterima dengan baik dan masuk kedalam jaringan firebase. Keluaran *packet loss* tersebut termasuk sangat baik menurut standar TIPHON.

Pada pengujian *delay* menghasilkan keluaran *delay* yang terbaca pada wireshark dari rata-rata *delay* yang sudah dilakukan pengujian 10 kali menghasilkan *delay* 245,3492 ms. Keluaran *delay* tersebut termasuk baik menurut standar TIPHON.

Pada pengujian *jitter* menghasilkan keluaran *jitter* yang terbaca pada wireshark dari rata-rata *jitter* yang sudah dilakukan pengujian sebanyak 10 kali menghasilkan *jitter* sebesar 236,6453 ms. Keluaran *jitter* tersebut termasuk kategori buruk menurut standar TIPHON.

V. KESIMPULAN

Pada pengujian aplikasi terbagi menjadi 2 bagian, yaitu pengujian kualitas jaringan pada saat pengiriman data kefirebase dan pengujian kualitas jaringan pada saat pengiriman data kewhatsapp. Berdasarkan hasil pengujian yang telah dilakukan memberikan kesimpulan bahwa:

- Pengujian kualitas jaringan pada saat pengiriman data kefirebase mendapatkan hasil *throughput* sebesar 15,635 Kbps. Kemungkinan *Throughput* yang didapat buruk dikarenakan kualitas jaringan yang buruk, dikarenakan pengujian dilakukan pada saat jam sibuk. Hasil dari *Packet loss* yang dihasilkan pada pengujian pengiriman data kefirebase yaitu 0%. Hasil ini dapat disimpulkan bahwa semua paket dapat terkirim dengan baik. Hasil dari *Delay* yang dihasilkan pada pengujian pengiriman data kefirebase sebesar 87,76367 ms. *Delay* yang didapatkan tidak nol berpengaruh juga oleh kualitas jaringan yang digunakan. Akan tetapi *delay* yang didapatkan pada pengujian kali ini termasuk kategori

sangat baik. Hasil *jitter* yang dihasilkan pada pengujian pengiriman kefirebase sebesar 73,82095 ms. *Jitter* yang didapatkan termasuk kekategori baik menurut standarisasi TIPHON.

- Pengujian kualitas jaringan pada saat pengiriman data kewhatsapp mendapatkan hasil *throughput* sebesar 3,9293 Kbps. Kemungkinan *Throughput* yang didapat buruk dikarenakan kualitas jaringan yang buruk, dikarenakan pengujian dilakukan pada saat jam sibuk. Hasil dari *Packet loss* yang dihasilkan pada pengujian pengiriman data kewhatsapp yaitu 0%. Hasil ini dapat disimpulkan bahwa semua paket dapat terkirim dengan baik. Hasil dari *Delay* yang dihasilkan pada pengujian pengiriman data kewhatsapp sebesar 245,3492 ms. *Delay* yang didapatkan tidak nol berpengaruh juga oleh kualitas jaringan yang digunakan. Akan tetapi *delay* yang didapatkan pada pengujian kali ini termasuk kekategori baik. Hasil *jitter* yang dihasilkan pada pengujian pengiriman kewhatsapp sebesar 236,6453 ms. *Jitter* yang didapatkan *Jitter* buruk karena prioritas yang tidak menguntungkan antara informasi sensitif waktu dan pergerakan informasi besar.

UCAPAN TERIMA KASIH

Terimakasih kepada semua pihak yang telah membantu tenaga ataupun pikiran untuk membuat jurnal ini. Terimakasih kepada kedua orangtua karena telah memberikan kasih sayang hingga saat ini.

DAFTAR PUSTAKA

- [1] R. J. Kodoatie, *Rekayasa dan Manajemen banjir kota*, Cet.1. Yogyakarta: Andi, 2013. [Online]. Available: https://perpustakaan.bnpb.go.id/bulian/index.php?p=show_detail&id=169
- [2] A. Findayani, "Kesiapan Siaga Masyarakat Dalam Penanggulangan Banjir Di Kota Semarang," *J. Geogr. Media Inf. Pengemb. dan Profesi Kegeografian*, vol. 12, no. 1, pp. 102–114, 2015.
- [3] Y. Saragih, J. H. Prima Silaban, H. Aliya Roostiani, and A. S. Elisabet, "Design of Automatic Water Flood Control and Monitoring Systems in Reservoirs Based on Internet of Things (IoT)," *Mecn. 2020 - Int. Conf. Mech. Electron. Comput. Ind. Technol.*, pp. 30–35, 2020, doi: 10.1109/MECNIT48290.2020.9166593.
- [4] G. Wang, C. Gu, J. Rice, T. Inoue, and C. Li, "Highly accurate noncontact water level monitoring using continuous-wave Doppler radar," *WiSNet 2013 - Proc. 2013 IEEE Top. Conf. Wirel. Sensors Sens. Networks - 2013 IEEE Radio Wirel. Week, RWW 2013*, no. January, pp. 19–21, 2013, doi: 10.1109/WiSNet.2013.6488620.
- [5] A. K. M. Riny Sulistyowati, Hari Agus Sujono, "Sistem Pendeteksi Banjir Berbasis Sensor Ultrasonik Dan Mikrokontroler," *Semin. Nas. Sains dan Teknol. Terap.*, no. January, pp. 49–58, 2015, [Online]. Available: <https://jurnal.itats.ac.id/sistem-pendeteksi-banjir-berbasis-sensor->

- ultrasonik-dan-mikrokontroler-dengan-media-komunikasi-sms-gate-way/
- [6] M. Khan, B. N. Silva, and K. Han, "Internet of Things Based Energy Aware Smart Home Control System," *IEEE Access*, vol. 4, pp. 7556–7566, 2016, doi: 10.1109/ACCESS.2016.2621752.
 - [7] M. R. Fahlevi and H. Gunawan, "Perancangan Sistem Pendeteksi Banjir Berbasis Internet of Things," *It (Informatic Tech. J.*, vol. 8, no. 1, p. 23, 2021, doi: 10.22303/it.8.1.2020.23-29.
 - [8] F. D. Hanggara, "Implementasi Internet of Things sebagai langkah mitigasi dini banjir," no. August 2017, pp. 2–3, 2020.
 - [9] Kuswindarini and dkk, "Sistem Pengendali Kebocoran Lpg Dengan Media Komunikasi Instant Messaging Whatsapp Berbasis Internet of Things," vol. 7, no. 2, pp. 4219–4226, 2019.
 - [10] P. Sindu Prawito and H. P. Perdana, "Aplikasi Sistem Manajemen Belajar Berbasis Web Dengan Framework Laravel di Growth2tech," *Syntax Lit. ; J. Ilm. Indones.*, vol. 5, no. 3, p. 100, 2020, doi: 10.36418/syntax-literate.v5i3.981.