

# IMPLEMENTASI RESERVOIR UNTUK PENANGANAN BANJIR DENGAN SENSOR KETINGGIAN AIR BERBASIS IOT (IMPLEMENTATION OF RESERVOIR FOR FLOOD MANAGING MANAGEMENT WITH WATER LEVEL SENSOR WITH IOT)

Muhammad Belva Al Kautsar Purwita<sup>1</sup>, Favian Dewanta<sup>2</sup>, Fardan<sup>3</sup>

<sup>1,2,3</sup>Universitas Telkom, Bandung

mbelvaalkautsar@student.telkomuniversity.ac.id<sup>1</sup>, favian@telkomuniversity.ac.id<sup>2</sup>,  
fardanfnn@telkomuniversity.ac.id<sup>3</sup>

## Abstrak

Bencana banjir sering melanda perumahan yang ada di Indonesia. Dampak dari bencana banjir tersebut tidak dapat diminimalisir karena warga setempat tidak memiliki kemampuan untuk mendapatkan informasi ketinggian air secara real-time. Selain itu mekanisme bendungan dan saluran air juga tidak efektif dalam penanganan banjir. Untuk menangani masalah tersebut maka pompa air dan tabung reservoir berbasis IoT dapat dijadikan alternatif dalam proses penanganan banjir yang terjadi di lingkungan perumahan. Dengan perkembangan teknologi yang pesat maka ada cara untuk menanggulangi bencana banjir dan meminimalisir kerugian yang terjadi akibat bencana banjir. Karena sistematis bendungan dan saluran air biasa kurang efektif untuk proses penanganan banjir di perumahan, maka pompa air dan tabung reservoir berbasis IoT dapat dijadikan alternatif dalam proses penanganan banjir yang terjadi di lingkungan perumahan. Tugas akhir ini telah membuat alat untuk penanganan banjir di perumahan dengan menggunakan alat yang sudah terhubung dengan sensor ultrasonik yang juga sudah disambungkan dengan NodeMCU ESP8266 dan mesin penyedot air yang dipasang di dataran terendah di sebuah perumahan dan disambungkan kepada tabung reservoir serta saluran air dengan pompa penyedot air. Untuk mengamati ketinggian air, penghuni dapat memantau melalui aplikasi selular yang juga sudah dilengkapi dengan fitur pemantauan tingkat ketinggian air. Ketika terjadi banjir, *buzzer* yang dipasangkan pada alat pendeteksi banjir akan berbunyi sesuai tingkat status ketinggian banjir. Jika penghuni ingin melihat catatan banjir yang terjadi di waktu sebelumnya, penghuni dapat melihatnya di *database* pengelola perumahan yang sudah disediakan.

**Kata Kunci :** Tabung Reservoir, IoT, Aplikasi selular, *Database*, Pompa penyedot air, Sensor ultrasonik.

## Abstract

*Flood disasters often hit existing housing in Indonesia. The impact of the flood disaster cannot be minimized because local residents do not have the ability to obtain real-time water level information. In addition, dam mechanisms and waterways are also ineffective in flood management. To address this problem, IoT-based water pumps and reservoir tubes can be used as an alternative in the process of handling floods that occur in residential environments. This research has created a flood management tool in housing which consists of an ultrasonic sensor that has been connected to the ESP8266 NodeMCU and a water suction machine that is placed in the lowest area in the residential area. The device is connected to a reservoir tube and a water line with a water suction pump. The process of observing water levels can be done through a mobile application that is equipped with a water level monitoring feature. In the event of a flood, the buzzer installed on the flood detection device will sound according to the status of the flood height. Residents of housing can also view flood records that have occurred in the past through the application features for housing managers that have been provided.*

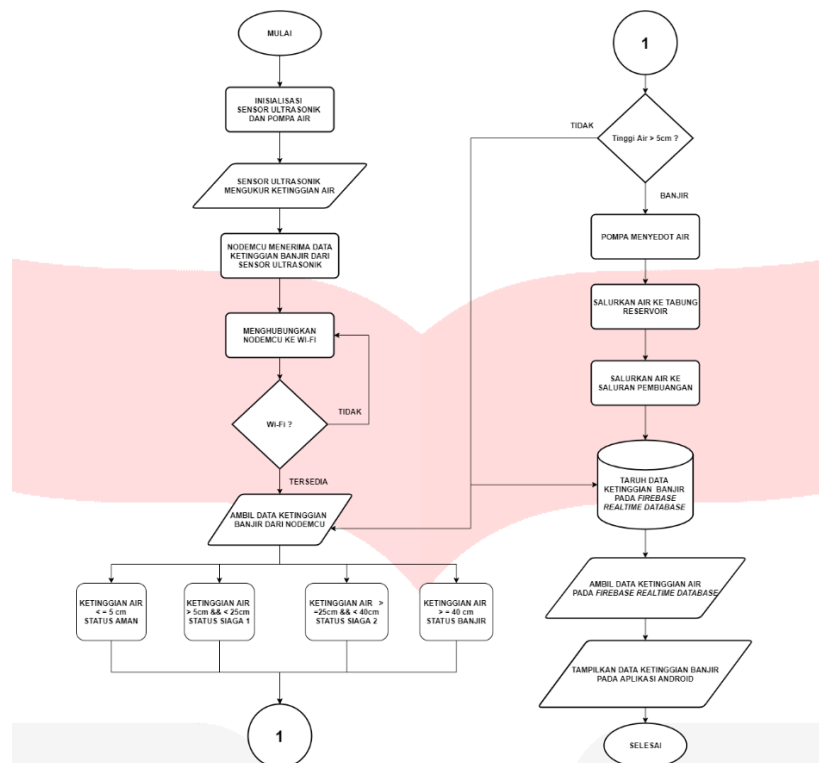
**Keywords:** Reservoir Tube, IoT, Mobile Application, *Database*, Suction pump, Ultrasonic sensor.

## 1. Pendahuluan

Permasalahan banjir di Indonesia ini sudah menjadi hal yang biasa terjadi dan berulang setiap tahunnya. Pada daerah perumahan, musim hujan yang berkepanjangan akibat perubahan cuaca ekstrem yang terjadi di berbagai wilayah di Indonesia menyebabkan masyarakat harus lebih waspada terhadap bencana banjir [1]. Bencana banjir selalu memakan korban baik materi sekalipun nyawa. Untuk meminimalkan dampak bencana banjir, diperlukan suatu terobosan untuk membantu menanggulangi banjir yang ada di Indonesia dengan membuat sistem penarikan air menggunakan reservoir serta pompa yang diletakkan di setiap perumahan yang ada di daerah banjir. Sudah ada yang pernah menggunakan manajemen reservoir loop tertutup untuk asimilasi data yang merupakan kenaikan paling curam untuk optimasi pemantauan berdasarkan koordinasi *multicriteria* [2]. Pada tugas akhir sebelumnya dicoba dengan menggunakan Sistem pemantauan ketinggian permukaan air dengan mengimplementasikan sensor ultrasonik berbasis mikrokontroler Arduino uno yang sudah terintegrasi pada sistem peringatan dini berbasis sms, yang akan mengetahui ketinggian permukaan air yang dibuat pada level-level tertentu [3]. Pada tugas akhir ini telah dibuat alat penanggulangan banjir berbasis *IoT* dengan menggunakan tabung reservoir dan pompa air yang sudah dihubungkan dengan sensor ultrasonik berbasis ESP8266 yang sudah diintegrasikan dengan *database firebase* secara *realtime* dan dapat di kendalikan melalui aplikasi selular berbasis android dengan fitur ketinggian air.



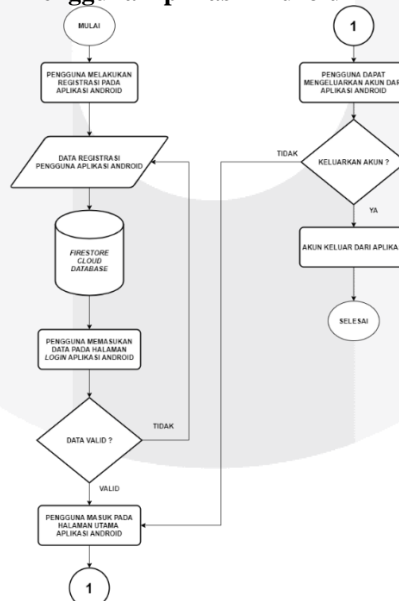
### 2.3 Diagram Alir Pengujian Sistem



**Gambar 3** Diagram Alir Pengujian Sistem.

**Gambar 3** memperlihatkan *flowchart* pengujian sistem yang dijelaskan sebagai berikut. Prosesnya sama dengan **Gambar 2** yang berbeda hanya ada parameter tinggi banjir yaitu aman pada ketinggian  $\leq 5$ cm, siaga 1 pada ketinggian  $> 5$ cm hingga  $< 25$ cm, siaga 2 pada ketinggian  $\leq 25$ cm hingga  $< 40$ cm, banjir pada ketinggian  $\geq 40$ cm. Ketika ketinggian air melewati batas aman yaitu  $> 5$ cm, pompa akan otomatis menyala dan menyedot air banjir dan langsung menyalurkannya melalui tabung reservoir.

### 2.4 Diagram Alir Pengelolaan Akun Pengguna Aplikasi Android



**Gambar 4** Diagram Alir Pengelolaan Akun Aplikasi Android.

**Gambar 4** memperlihatkan pengguna aplikasi reservoir wajib melakukan registrasi akun terlebih dahulu pada halaman registrasi, lalu data tersebut akan langsung dikirimkan kepada *firebase firestore database*. Setelah pengguna melakukan registrasi, pengguna dapat melakukan aktivitas *login* pada halaman *login*. Ketika pengguna memasukkan data pada halaman *login*, aplikasi reservoir akan mengecek keaslian akun dengan cara menyamakan data *login* dengan data yang ada pada *firebase database*, ketika data sudah sesuai, pengguna dapat masuk ke dalam aplikasi. Ketika data tidak sesuai maka pengguna tidak dapat masuk dan harus memperbaiki data login terlebih dahulu. Pengguna aplikasi juga dapat mengeluarkan akun pada aplikasi Reservoir.

## 2.5 Realisasi Perangkat Keras

Realisasi perangkat keras yang dirancang pada tugas akhir ini menggunakan bahan dasar akrilik yang diisi dengan alat serta sensor ultrasonik untuk membaca ketinggian air pada saat terjadi banjir.



**Gambar 5** Realisasi Perangkat Keras.

**Gambar 5** merupakan prototipe berupa kotak akrilik dengan spesifikasi panjang 15cm, lebar 10cm, dan tinggi 45 cm sebagai media untuk melakukan simulasi pengukuran tinggi air banjir.

## 3. HASIL PENGUJIAN DAN ANALISIS

### 3.1 Pengujian Pada Alat

Pengujian pada alat dilakukan untuk mengetahui apakah kinerja alat dan sensor berfungsi sesuai dari yang direncanakan dan dapat berfungsi sesuai yang diinginkan. Berikut hasil dari pengujian alat serta sensor.

**Tabel 2** Hasil Uji Coba Alat.

No	Nama Alat & Sensor	Fungsi	Hasil
1	NodeMCU ESP8266 Wi-Fi	Menghubungkan sensor, alat, jaringan Wi-Fi, dan database <i>firebase</i> .	Berhasil
2	Sensor Ultrasonik	Membaca Ketinggian Banjir Sesuai Tingkat Status Banjir dan menyalurkannya kepada ESP8266.	Berhasil
3	Relay	Menyambungkan dan memutus arus pada pompa penyedot air serta terhubung pada ESP8266.	Berhasil
4	Pompa Penyedot Air	Menyala dan mati sesuai perintah dan kondisi air.	Berhasil

5	Power Supply 12 Volt	Menghantarkan arus listrik kepada <i>relay</i> .	Berhasil
6	Tabung Reservoir	Menjadi tempat penyimpanan dan menjadi jalur air.	Berhasil
7	Buzzer	Berbunyi sesuai dengan ketinggian air.	Berhasil

### 3.2 Hasil Pengujian Kompatibilitas Aplikasi Reservoir

Pengujian kompatibilitas tampilan aplikasi Reservoir pada tugas akhir ini dilakukan dengan 2 cara yaitu dengan smart phone dan metode robotest. Hasil pengujian ditunjukkan dalam tabel berikut.

**Tabel 3** Hasil Pengujian UI/UX Pada Smart Phone.

No	Nama Perangkat	Versi Android / API	Ukuran Layar (Inci)	Resolusi Layar (Pixels)	Hasil
1	Redmi Note 9 Pro	10 / 29	6.67"	1080 x 2400	Berhasil
2	Samsung Galaxy S8+	9 / 28	6.62"	1080 x 1920	Berhasil
3	Oppo F3 Plus	6.0.1 / 23	6"	1080 x 1920	Berhasil
4	Realme XT	10/29	6.4"	1080 x 2340	Berhasil
5	Asus Zenfone Max Pro M2	9/28	6.26"	1520 x 720	Berhasil
6	Samsung S10Lite 2020	11/33	6.7"	1080 x 2400	Berhasil

**Tabel 4** Hasil Pengujian Pada Simulasi Robotest Firebase Test Lab.

No	Nama Perangkat	Versi Android / API	Ukuran Layar	Resolusi Layar (Pixels)	Hasil
1	Nokia 9	9 / 28	6"	1440 x 2880	Berhasil
2	Vivo 1805	8.1 / 27	6.59"	1080 x 2316	Berhasil
3	Samsung SM-G891A	8.0 / 26	5.1"	2560 x 1440	Berhasil
4	LG-X230	6.0 / 23	3"	854x480	Berhasil
5	Moto G Play	6.0 / 23	6.5"	720 x 1600	Berhasil

**Tabel 3 dan 4** merupakan hasil pengujian aplikasi yang dilakukan menggunakan simulator yang ada pada *firebase test lab* dan menggunakan ponsel android dengan tipe dan spesifikasi android yang berbeda.

### 3.3 Hasil Pengujian Dengan 4 Status Ketinggian Air

**Tabel 5** Hasil Pengujian 4 Status Ketinggian Air

No	Status Air	Tinggi Air	Kondisi Buzzer	Pompa Permukaan	Pompa Reservoir	Hasil
1	Aman	<= 5cm	Mati	Mati	Mati	Berhasil
2	Siaga 1	> 5cm && < 25 cm	Menyala	Menyala	Menyala	Berhasil
3	Siaga 2	>= 25cm && < 40 cm	Menyala	Menyala	Menyala	Berhasil
4	Banjir	>= 40 cm	Menyala	Menyala	Menyala	Berhasil

### 3.4 Hasil Pengujian Kualitas Jaringan

Pengujian kualitas jaringan dilakukan untuk mengetahui kualitas dari jaringan yang digunakan dalam menjalankan suatu sistem. Pengujian kualitas jaringan pada tugas akhir ini akan menghitung parameter nilai *delay*, *throughput*, dan *packet loss*. Hasil dan cara pengujian kualitas jaringan akan ditampilkan pada gambar berikut.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.236	35.201.97.85	TLSv1.2	456	Application Data
3	0.023082	192.168.137.236	35.201.97.85	TLSv1.2	77	Application Data
6	0.323269	192.168.137.236	35.201.97.85	TLSv1.2	417	Application Data
8	0.346149	192.168.137.236	35.201.97.85	TLSv1.2	77	Application Data
11	0.615395	192.168.137.236	35.201.97.85	TCP	54	56267 → 443 [ACK] Seq=812 Ack=805 Win=2144 Len=0
12	0.729889	192.168.137.236	35.201.97.85	TLSv1.2	441	Application Data
14	0.753020	192.168.137.236	35.201.97.85	TLSv1.2	84	Application Data
17	1.034387	192.168.137.236	35.201.97.85	TLSv1.2	441	Application Data
19	1.061441	192.168.137.236	35.201.97.85	TLSv1.2	84	Application Data
22	1.327559	192.168.137.236	35.201.97.85	TCP	54	56267 → 443 [ACK] Seq=1646 Ack=1593 Win=2144 Len=0
23	1.341275	192.168.137.236	35.201.97.85	TLSv1.2	432	Application Data

**Gambar 6** Filter TCP Serta IP Sumber dan Tujuan.

**Gambar 6** memperlihatkan cara menyaring IP sumber dan tujuan serta protokol yang ada pada hasil pengukuran *delay*. *Filter* yang digunakan pada tugas akhir ini adalah penyaringan IP NodeMCU sebagai sumber



dan IP *firebase database* sebagai tujuan dengan menggunakan protokol tcp. IP dari NodeMCU ESP8266 yang didapatkan adalah 192.168.137.236 dan IP dari *firebase* adalah 35.201.97.85.

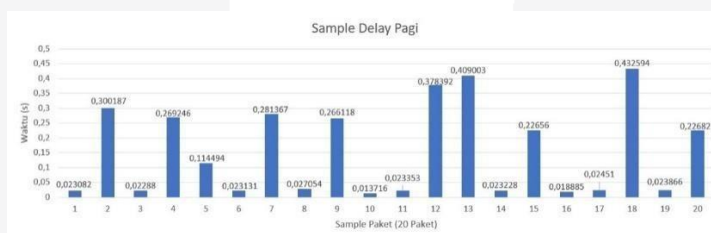
No.	Time	Source	Destination	Protocol	Length	Time delta info	Time delta from previous displayed frame
1	0.000000	192.168.137.236	35.201.97.85	TLSv1.2	456	0.000000 Application Data	0
3	0.023082	192.168.137.236	35.201.97.85	TLSv1.2	77	0.023082 Application Data	0,02308
6	0.322369	192.168.137.236	35.201.97.85	TLSv1.2	417	0.300187 Application Data	0,30019
8	0.346149	192.168.137.236	35.201.97.85	TLSv1.2	77	0.022880 Application Data	0,02288
11	0.615395	192.168.137.236	35.201.97.85	TCP	54	0.269246 Seq=6267 > 443 [ACK] Seq=812 Ack=805 Win=2144 Len=0	0,26925
12	0.779889	192.168.137.236	35.201.97.85	TLSv1.2	441	0.114494 Application Data	0,11449
14	0.753020	192.168.137.236	35.201.97.85	TLSv1.2	84	0.023131 Application Data	0,02313
17	1.034387	192.168.137.236	35.201.97.85	TLSv1.2	441	0.281367 Application Data	0,28137
19	1.061441	192.168.137.236	35.201.97.85	TLSv1.2	84	0.027054 Application Data	0,02705
22	1.327559	192.168.137.236	35.201.97.85	TCP	54	0.266118 Seq=6267 > 443 [ACK] Seq=1646 Ack=1593 Win=2144 Len=0	0,26612
23	1.341275	192.168.137.236	35.201.97.85	TLSv1.2	432	0.013716 Application Data	0,01372
25	1.364628	192.168.137.236	35.201.97.85	TLSv1.2	83	0.023353 Application Data	0,02335
29	1.743020	192.168.137.236	35.201.97.85	TCP	54	0.378392 Seq=6267 > 443 [ACK] Seq=2053 Ack=1986 Win=1751 Len=0	0,37839
30	2.152023	192.168.137.236	35.201.97.85	TLSv1.2	456	0.409003 Application Data	0,409
32	2.175251	192.168.137.236	35.201.97.85	TLSv1.2	77	0.023228 Application Data	0,02323
35	2.401811	192.168.137.236	35.201.97.85	TCP	54	0.226560 Seq=6267 > 443 [ACK] Seq=2478 Ack=2403 Win=2144 Len=0	0,22656
36	2.420696	192.168.137.236	35.201.97.85	TLSv1.2	417	0.018885 Application Data	0,01889
38	2.445206	192.168.137.236	35.201.97.85	TLSv1.2	77	0.024510 Application Data	0,02451 Sample Pagi
41	2.877800	192.168.137.236	35.201.97.85	TLSv1.2	441	0.432594 Application Data	0,43259
43	2.901666	192.168.137.236	35.201.97.85	TLSv1.2	84	0.023866 Application Data	0,02387 Total Sample Delay
46	3.128487	192.168.137.236	35.201.97.85	TCP	54	0.226821 Seq=6267 > 443 [ACK] Seq=3281 Ack=3184 Win=2144 Len=0	0,22682 3,128487 0,15642435

**Gambar 7** Contoh Perhitungan Delay Pada Ms.Excel

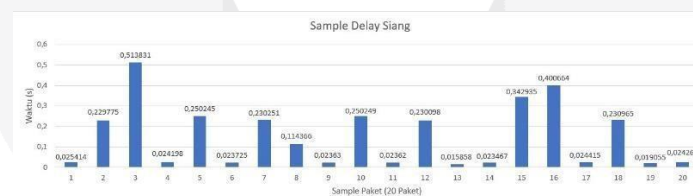
**Gambar 7** merupakan hasil dari format Ms.Excel dari hasil perhitungan *delay* pada aplikasi *wireshark*. Pada Ms.Excel diambil contoh hasil *delay* sebanyak 20 paket dan diambil perhitungan waktunya. Perhitungan Waktu selisih pengiriman data tertera pada tabel *time delay info* lalu akan dibuat perhitungan waktu menjadi satuan detik. Setelah membuat satuan waktu menjadi satuan detik, 20 paket contoh perhitungan *delay* akan di total untuk mencari rata rata *delay* pada masing masing zona waktu percobaan pengambilan *delay*. Cara mencari rata rata *delay* pada tugas akhir ini adalah dengan cara membagi hasil total *delay* dengan 20 jumlah paket. Hasil perhitungan *delay* pada masing masing zona waktu percobaan akan dijelaskan pada sub-bab dan gambar berikut.

### 3.4.1 Hasil Perhitungan Delay

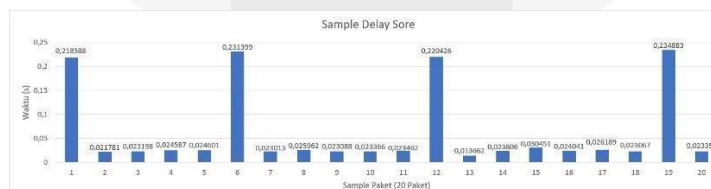
Hasil perhitungan nilai *delay* pada tugas akhir ini akan dibagi menjadi 4 waktu uji coba yaitu, pagi hari, siang hari, sore hari, dan malam hari. Uji coba sampel *delay* dalam tugas akhir ini menggunakan 20 sampel paket nilai *delay* yang sudah diambil dengan menggunakan aplikasi *wireshark* dengan cara menyaring protokol tcp dan menyaring IP dari ESP8266 dan *firebase database* pada saat melakukan skenario percobaan. Nilai *delay* yang diambil menggunakan satuan detik. Untuk hasil perhitungan nilai *delay* dari 4 waktu pengambilan percobaan akan dijelaskan pada gambar berikut.



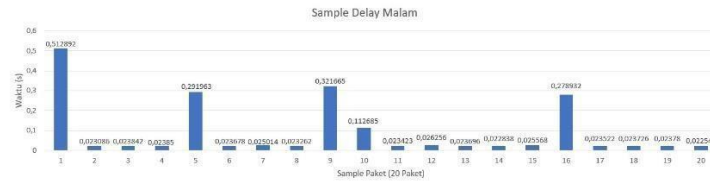
**Gambar 8** Grafik Delay Pagi Hari (06:51:26 - 06:56:34).



**Gambar 9** Grafik Delay Siang Hari (12:39:42 - 12:44:29).

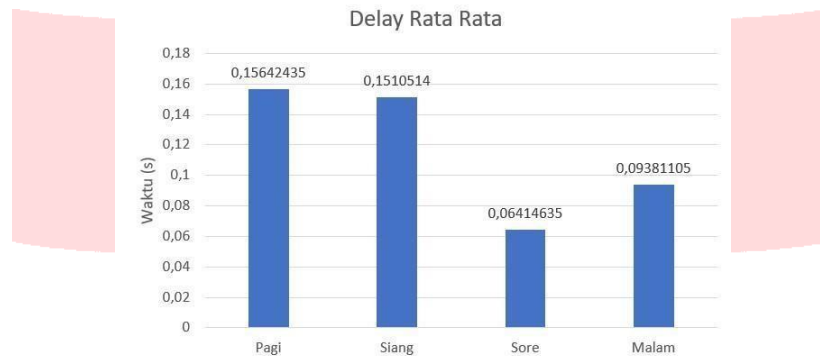


**Gambar 10** Grafik Delay Sore Hari (15:24:03 - 15:25:31).



**Gambar 11** Grafik Delay Malam Hari (00:42:47 - 00:46:51).

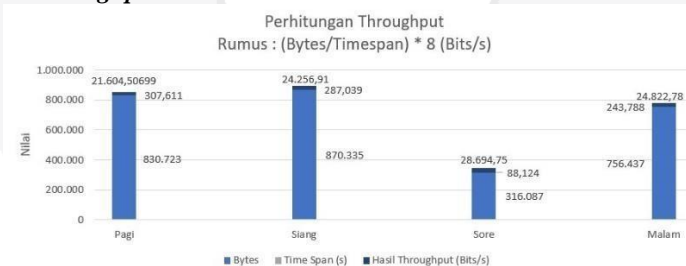
**Gambar 8** hingga **Gambar 11** merupakan hasil dari uji coba 20 paket nilai *delay* akan diambil nilai rata rata dari setiap waktu uji coba dengan cara menjumlahkan total nilai *delay* lalu membaginya dengan jumlah sampel paket. Hasil rata rata *delay* akan tertera pada gambar berikut.



**Gambar 12** Grafik Rata Rata Nilai Delay.

**Gambar 12** memperlihatkan hasil dari perhitungan nilai rata rata *delay* yang diambil dari 20 sampel data uji coba pengambilan nilai *delay* yang diambil pada 4 waktu yaitu pagi hari, siang hari, sore hari, dan malam hari. Hasil dari uji coba pengambilan nilai *delay* diambil dari 20 sampel paket pertama yang diambil oleh aplikasi *wireshark* pada saat uji coba pengiriman data ketinggian banjir oleh NodeMCU ESP8266 kepada *Firestore Database*. Skenario yang dilakukan pada pengujian *delay* ini menggunakan skenario pengujian pengosongan kotak akrilik yang dihitung pada aplikasi *wireshark* untuk menghitung jumlah paket beserta nilai *delay*. Paket yang didapatkan pada aplikasi *wireshark* dalam uji coba kali ini adalah mengukur *delay* dari pengiriman data oleh NodeMCU ESP8266 hingga data sampai kepada *firebase database*. Pengujian kali ini menggunakan metode penyurutan banjir dari ketinggian  $\geq 40$  cm dengan status banjir hingga ketinggian 5cm dengan status aman. Pada saat proses pengosongan, aplikasi *wireshark* dijalankan lalu dilakukan pengukuran kualitas *delay* pada sistem tugas akhir ini. Hasil dari rata rata *delay* pada pagi hari, siang hari, sore hari, dan malam hari sesuai dari uji coba pengambilan nilai *delay* yang dilakukan pada tugas akhir ini menghasilkan nilai *delay* yang sangat bagus yaitu dibawah 0,2 detik dan tergolong pada kategori *preferred* [4].

### 3.4.2 Hasil Perhitungan Throughput

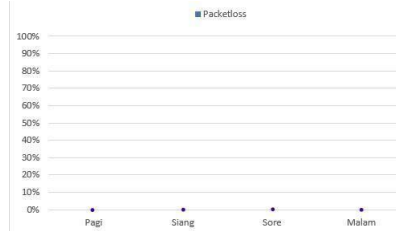


**Gambar 13** Hasil Perhitungan Throughput

**Gambar 13** memperlihatkan hasil dari perhitungan *throughput* yang dilakukan pada aplikasi *wireshark* yang diambil pada 4 jenis waktu yaitu pada pagi hari, siang hari, sore hari, dan malam hari. Nilai *throughput* ini menghasilkan nilai yang berbeda beda pada setiap waktunya. Nilai *throughput* pada tugas akhir ini ditetapkan pada satuan bits/s.

Nilai *throughput* pada percobaan pagi hari dihasilkan nilai *throughput* sebesar 21 Kbps, pada siang hari dihasilkan nilai *throughput* sebesar 24 Kbps, pada sore hari dihasilkan nilai *throughput* sebesar 28 Kbps dan pada malam hari dihasilkan nilai *throughput* sebesar 24 Kbps. Hal tersebut dapat disimpulkan bahwa nilai *throughput* yang didapatkan sesuai dari jumlah aktivitas data yang dilakukan pada jaringan tersebut dan tergantung pada lama percobaan dilakukan. Nilai *throughput* yang terbesar adalah nilai *throughput* pada sore hari, dikarenakan memiliki waktu pengujian yang paling cepat yaitu sekitar 1 menit. Nilai *throughput* dapat dipengaruhi oleh jumlah aktivitas dan lama waktu dari sebuah aktivitas pengiriman data.

### 3.4.3 Hasil Perhitungan Packet Loss



**Gambar 14** Hasil Pengukuran Packetloss.

**Gambar 14** memperlihatkan hasil dari pengukuran nilai packetloss pada tugas akhir ini. Nilai packetloss yang dihasilkan pada pagi, siang, sore, dan malam hari adalah 0% dan tidak ada paket yang hilang atau tidak terkirim pada tugas akhir ini. Nilai packetloss dapat dilihat dari jumlah perbandingan paket yang dikirimkan dengan jumlah paket yang diterima. Menurut ITU-T G.1010 *web browsing* nilai *packetloss* yang dihasilkan pada tugas akhir ini merupakan nilai yang sangat bagus atau masuk dalam kategori *preferred* yaitu dengan nilai 0.

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

Berdasarkan hasil dari pengujian dan analisa pada tugas akhir ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Integrasi sensor ultrasonik dengan NodeMCU ESP8266 untuk membaca ketinggian banjir dapat terhubung dengan baik dan dapat membaca ketinggian permukaan air dengan baik sesuai dengan pengukuran pada penggaris.
2. Aplikasi Reservoir dapat mengambil nilai ketinggian air banjir secara langsung dengan nilai *delay* yang sangat kecil, dan NodeMCU ESP8266 merupakan mikrokontroler yang praktis.
3. Hasil dari *delay* pengiriman nilai ketinggian air oleh sensor ultrasonik ke *database firebase* merupakan nilai *delay* yang sangat bagus yaitu dengan nilai  $< 0,1$  detik pada setiap paketnya, maka *database firebase* memiliki keakuratan yang baik untuk melakukan transfer data secara *realtime*.
4. Aplikasi android studio dapat membuat aplikasi android secara bebas membuat tampilan dengan menggunakan sistematis xml dan logika bahasa pemrograman java dan dapat terintegrasi dengan *database firebase*.
5. Aplikasi Reservoir dapat menaruh data akun pengguna aplikasi pada *firebase firestore cloud database* untuk mencatat data pengguna aplikasi dan dapat dilihat oleh pemegang akun *database firebase*.
6. Nilai *throughput* yang diperoleh dari tugas akhir ini adalah pagi hari sebesar 21 Kbps, siang hari sebesar 24 Kbps, sore hari sebesar 28 Kbps, dan malam hari sebesar 24 Kbps. Nilai *throughput* maksimal didapatkan dari kondisi jaringan yang memiliki *traffic* yang minimum.
7. Nilai *packetloss* yang diperoleh adalah 0. Semakin sedikit nilai *packetloss* maka akan lebih baik juga kualitas dari suatu jaringan karena semakin sedikit paket yang hilang.

### 4.2 Saran

1. Berdasarkan hasil dari uji coba dalam tugas akhir ini maka ditemukan beberapa kekurangan yang dapat dikembangkan dari tugas akhir ini, yaitu :
2. Pengembang selanjutnya dapat menampilkan fitur catatan dan notifikasi kejadian banjir melalui aplikasi android.
3. Diharapkan sensor yang digunakan pada tugas akhir ini dapat ditambah agar mendukung fitur yang akan ditampilkan pada aplikasi android.
4. Sebaiknya dibuat aplikasi selular untuk *ios / apple iphone*.
5. Sebaiknya kotak akrilik pengukuran tinggi banjir dibuat dalam ukuran yang lebih tinggi dan lebih lebar agar memudahkan untuk menaruh pompa serta memudahkan untuk sensor ultrasonik membaca ketinggian air tanpa ada penghalang.

## REFERENSI

- [1] Wahyono, Irawan Dwi, Asfani, Khoirudin, Aripriharta, Fadlika, Ilham, "New Method of Artificial Intelligence For Disaster Information Floods Use Distributed Wireless Sensor," 2019 International Seminar on Application for Technology of Information and Communication (iSemantic). IEEE, 2019.
- [2] Skulimowski, Andrzej M.J, "Multicriteria Coordination of Flood Control In Water Reservoir Systems," IEEE Robotics and Automation Society and Institute of Electrical and Electronics Engineers, 2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR).
- [3] W. Indianto and A. Harsa Kridalaksana, "Perancangan Sistem Prototipe Pendeteksi Banjir Peringatan Dini Menggunakan Arduino dan PHP," 8. Wahyu Indianto (Awang, Yuli) Hal 45-49," *Jurnal Informatika Mulawarman*, vol. 12, no. 1, p. 45, 2017.
- [4] "ITU-T End-user multimedia QoS categories," 2001.