

Implementasi Reservoir Untuk Penanganan Bencana Banjir Berbasis *Internet Of Things* (*Reservoir Implementation For Flood Handling Based On Internet Of Things*)

1st Firman Rizkianoor
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
firmanrizkianoor@student.telkomu
niversity.ac.id

2nd Favian Dewant
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
favian@telkomuniversity.ac.id

3rd Fardan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
fardanfn@telkomuniversity.ac.id

Abstrak

Penelitian ini didasarkan kepada terjadinya bencana banjir di Indonesia yang kerap kali menjadi permasalahan masyarakat banyak di Indonesia. Banjir merupakan salah satu bencana alam yang sering melanda Indonesia beberapa tahun terakhir. Bencana banjir juga telah menjadi sorotan oleh pemerintah untuk di tangani secara sigap. Hal ini terlihat dari peran pemerintah dalam pembentukan Badan Penanggulangan Bencana Nasional (BPBN) yang berperan sebagai badan dari pemerintah dalam penanggulangan bencana di Indonesia. Untuk itu penulis merancang sebuah purwarupa yang dapat di implementasikan dari sektor perumahan untuk percobaan langkah awal purwarupa tersebut. Penulis mengembangkan sistem penanganan banjir dengan reservoir berbasis IoT dengan adanya alat ini di harapkan dapat mengantisipasi banjir dari suatu daerah perumahan dan juga dapat memberikan informasi tentang situasi banjir yang terjadi di lingkungan tersebut. Perancangan alat ini disertai dengan hasil pengujian yang cukup memuaskan. Alat yang dirancang dapat bekerja secara optimal, *website* yang dibuat juga dapat diakses dan menampilkan informasi yang dibutuhkan. Pengujian QoS baik untuk sistem dan akses *website* juga memiliki hasil yang bagus. Pada pengujian sistem dengan *wireshark*, diperoleh rata-rata *delay* sebesar 0.17959 s, *packet loss* sebesar 0%, serta *throughput* sebesar 50.94 Kb/s. Pada pengujian *website* dengan *Apache JMeter*, diperoleh rata-rata *delay* sebesar 0.307413s dan *throughput* sebesar 30.96 Kb/s.

Kata kunci : Banjir, IoT, NodeMcu, MySQL, Arduino IDE, VPS

Abstract

This research is based on the occurrence of floods in Indonesia which is often a problem for many people in Indonesia. Flood is one of the natural disasters that often hit Indonesia in recent years. The flood disaster has also been in the

*spotlight by the government to be handled swiftly. This can be seen from the government's role in the establishment of the National Disaster Management Agency (BPBN) which acts as a government agency in disaster management in Indonesia. For this reason, the authors designed a prototype that could be implemented from the housing sector for the initial trial of the prototype. The author develops a flood management system with an IoT-based reservoir with this tool in the hope that it can anticipate flooding from a residential area and can also provide information about the flood situation that occurs in the environment. This prototype also comes with an excellent result. The devices work exactly like the way they have to, and the website is accessible and shows the needed information. QoS testing to both system and website access also have good results. In testing the system with *wireshark*, the average delay is 0.17959 s, packet loss is 0%, and the throughput is 50.94 Kb/s. In testing the website with *Apache JMeter*, the average delay is 0.307413 s and the throughput is 30.96 Kb/s.*

Keywords: Flood, IoT, NodeMcu, MySQL, Arduino IDE, VPS

I. PENDAHULUAN

Banjir merupakan salah satu bencana alam yang sering melanda Indonesia beberapa tahun terakhir. Bencana banjir juga telah menjadi sorotan oleh pemerintah untuk ditangani secara sigap. Hal ini terlihat dari peran pemerintah dalam pembentukan Badan Penanggulangan Bencana Nasional (BPBN) yang berperan sebagai badan dari pemerintah dalam penanggulangan bencana di Indonesia[1]. Beberapa upaya yang telah dilakukan oleh pemerintah dalam menanggulangi banjir adalah seperti pembangunan bendungan, normalisasi sungai, dan tanggul laut raksasa. Adapun beberapa dampak positif dari upaya yang sudah dilakukan yaitu mengurangi resiko banjir, namun dampak negatif dari upaya tersebut adalah merelokasi pemukiman di bantaran sungai, dan anggaran biaya yang besar[2].

Salah satu implementasi dalam penanggulangan banjir berbasis IoT adalah IoT cloud data logger

guna mengantisipasi bencana banjir dengan menggunakan mikrokontroler Arduino Nano (ESP8266) dan sensor ultrasonik untuk mendeteksi ketinggian air. Hasil yang didapat dari penelitian tersebut adalah tercatatnya perubahan ketinggian dan percepatan air serta pengunggahan status melalui media sosial *Twitter* [3].

Sistem penanganan banjir dengan memanfaatkan sensor ketinggian air ini merupakan peralatan yang berfungsi mendeteksi ketinggian air untuk menjadi masukan pada mikrokontroler untuk memberikan pemberitahuan kepada orang-orang kondisi ketinggian air saat ini, peralatan tersebut terdiri dari sensor dan mikrokontroler [4].

Untuk penggunaan aplikasi yang diterapkan dengan mengusung konsep IoT yaitu konsep sensor ketinggian air yang berbasis *web* [5]. Terdapat banyak sensor yang dapat digunakan untuk mengetahui ketinggian permukaan air, pada topik kali ini sensor yang digunakan ialah sensor Ultrasonik yang diletakkan pada sebuah bidang atau tabung.

Penelitian ini diharapkan dapat diimplementasikan dengan menekan biaya agar dapat diterapkan di berbagai daerah kecil maupun besar serta menjadi solusi yang tepat untuk menanggulangi bencana banjir di daerah permukiman.

II. KAJIAN TEORI

a. Mikrokontroler

Mikrokontroler merupakan *Chip IC (Integrated Circuit)* yang dirancang untuk menerima sinyal *input*, memproses dan mengolahnya dan memberikan output sesuai dengan program yang dimasukkan. Dalam mikrokontroler terdapat *CPU, memory, jalur Input/output (I/O)* dan alat pendukung lainnya. Kecepatan operasi mikrokontroler pada umumnya di antara 1-16 MHz dan untuk dibandingkan dengan Komputer saat ini mikrokontroler jauh lebih kecil [6].

b. Arduino IDE

Arduino merupakan program *board* yang mempunyai bahasa sendiri menyerupai bahasa pemrograman C. Bahasa Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman. Di dalam IC Arduino terdapat suatu program yang tertanam bernama *Bootloader* sebagai penengah *compiler* antara Arduino dengan Mikrokontroler [6].

c. Node MCU ESP8266

Node MCU berjenis ESP8266 merupakan mikrokontroler yang pada umumnya difungsikan sebagai pembaca data yang didapat dari sensor, dan mengolahnya sehingga bisa dimengerti. Fungsi lain dari ESP8266 ialah sebagai pengirim data ke android.[6]

d. Sensor Ultrasonik

Sensor merupakan alat yang berfungsi untuk mengubah suatu nilai besaran fisik menjadi nilai besaran listrik agar dapat dianalisa dengan rangkaian

listrik tertentu. Terdapat sensor yang digunakan untuk membantu pengembangan teknologi masa kini. Terdapat beragam jenis sensor yang diketahui, sensor suhu, sensor *ultrasonic*, sensor suara, sensor *infra red*, sensor ketinggian air, dan beragam lainnya [4].

Sensor ultrasonik HC-SR04 merupakan sensor yang mengubah bentuk fisis (bunyi) menjadi bentuk satuan besaran listrik dan juga sebaliknya. Cara kerja sensor ultrasonik didasari dengan prinsip kerja dari pantulan suatu gelombang suara sehingga dapat digunakan menafsirkan atau memperkirakan jarak suatu benda dengan menggunakan frekuensi tertentu. Sensor ultrasonik menggunakan gelombang bunyi (*ultrasonic sound*) yang mempunyai spektrum gelombang yang sangat tinggi yaitu 20.000 Hz. [6]

e. Reservoir

Reservoir merupakan tempat penampungan air sementara yang dirancang untuk menampung air dalam jumlah tertentu agar dapat mengurangi kapasitas volume air di lain tempat. Reservoir yang digunakan yaitu *ground reservoir* untuk menampung air banjir dari suatu daerah, dan apabila reservoir telah terisi penuh maka reservoir memompa air ke aliran sungai.

Sistem dari *reservoir* sederhana terdiri dari :

- Pipa Inlet
- Pipa Outlet
- Pipa Peluap dan Penguras
- Pipa Udara
- Lubang Inspeksi
- Tangga naik ke reservoir
- Alat penunjuk tinggi muka air
- Alat pengukur debit air

f. Internet Of Things (IoT)

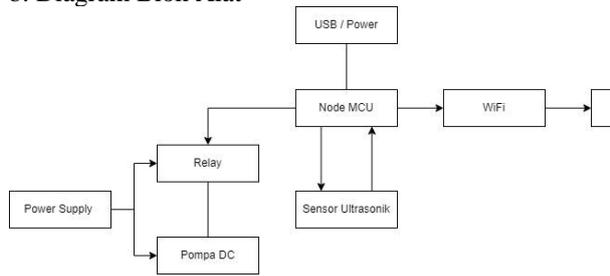
Internet of Things merupakan konsep dari komputasi / otomatisasi yang menuju kepada koneksi internet dan mengubah objek umum menjadi terhubung dengan internet. Dasar dari konsep IoT ini yaitu untuk meluncurkan jutaan bahkan miliaran objek pintar yang bisa mendeteksi keadaan lingkungan sekitarnya menerima dan mengolah datanya hingga mengirimkan data itu kembali kepada lingkungan itu sendiri [8].

g. VPS

VPS atau *Virtual Private Server* adalah sebuah teknologi untuk virtualisasi sebuah server. Pada dasarnya, sebuah *physical server* dibagi menjadi beberapa VPS sehingga terlihat dan bekerja seperti *server* mandiri. Tiap VPS memiliki Full Root Access, Operating System, dan pengaturan sendiri untuk script, user, processing, file system termasuk resource server. Biasanya menggunakan OS Linux. VPS tidak akan mempengaruhi VPS lain dalam sebuah server yang sama, berbeda dengan shared hosting.

III. METODE

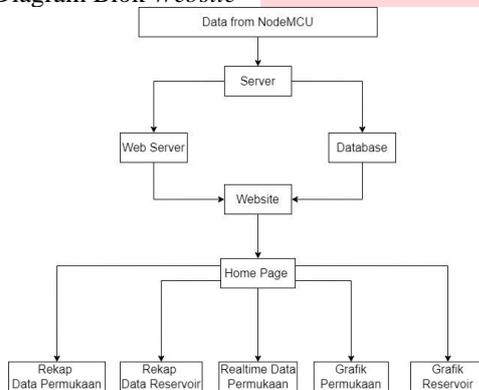
- a. Diagram Blok Perancangan Alat
- b. Diagram Blok Alat



Gambar 1 Diagram Blok Alat

Gambar 1 merupakan alur yang terjadi mulai dari alat menyala hingga data ditampilkan pada website.

- c. Diagram Blok Website



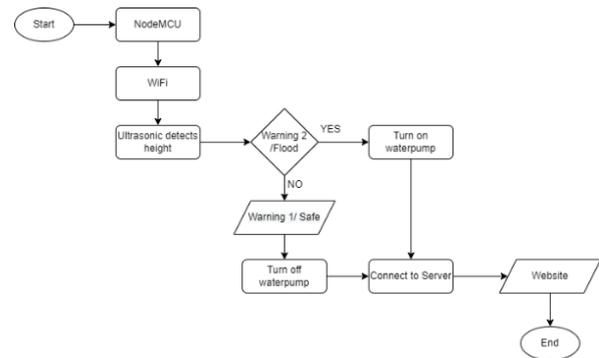
Gambar 2 Diagram Blok Website

Gambar 2 merupakan alur yang dilalui oleh data sensor sebelum pada akhirnya dapat dilihat pada website.

- d. Skenario Pengujian

Pengujian alat dapat dilakukan sesuai dengan langkah-langkah berikut yang dapat dilihat alurnya pada Gambar 3 di bawah :

- Langkah 1 : Menyalakan NodeMCU ESP8266
- Langkah 2 : Sensor Ultrasonik mendeteksi ketinggian permukaan air
- Langkah 3 : Data yang diperoleh sensor Ultrasonik kemudian dikonversi menjadi data PHP untuk dimasukkan ke database MYSQL
- Langkah 4 : Data yang tersimpan di database selanjutnya akan bisa diakses melalui web yang sudah dibuat.

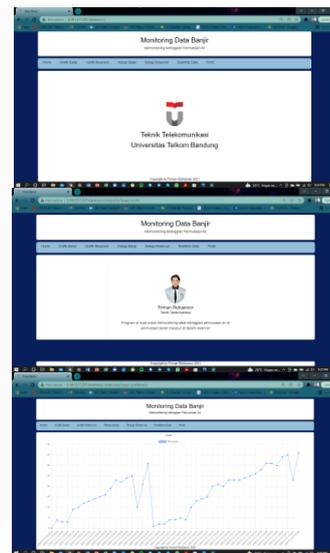


Gambar 3 Flowchart Sistem

IV. HASIL DAN PEMBAHASAN

- a. Pengujian Website
- b. Desain Website

Sebagai penunjang penulisan Penelitian ini, penulis membuat website sederhana menggunakan PHP, Bootstrap, CSS, dan Javascript. Website ini dapat menjadi panduan bagi pengguna untuk melihat data banjir, data reservoir, grafik ketinggian air, hingga realtime ketinggian air seperti pada Gambar 4. Website ini merupakan user interface yang terkoneksi dengan database dari MySQL dengan dibantu oleh VPS seperti yang dapat dilihat pada Gambar 5.



- (a).
- (b).
- (c).

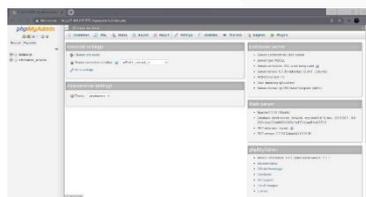


(d).

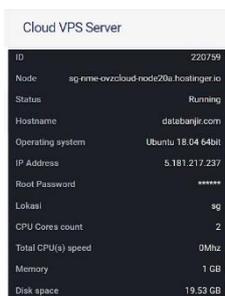


(f).

Gambar 4 Tampilan Website : (a). Halaman Awal (b). Profil Penulis (c). Grafik Banjir (d). Grafik Reservoir (e). Data Banjir (f). Data Reservoir (g). Realtime Data



(g).



(a).

(b).

Gambar 5 Penunjang Pembuatan Website : (a). MySQL (b). VPS

c. Pengujian Pada Alat

Pengujian pada alat dilakukan untuk memastikan alat yang digunakan sudah siap untuk diuji. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1 Hasil uji coba alat

No	Nama Alat & Sensor	Fungsi	Hasil
1	NodeMCU ESP8266-12E Wi-Fi	Menghubungkan Sensor Ultrasonik, jaringan Wi-Fi, dan Web	Dapat terhubung dengan jaringan Wi-Fi serta <i>mobile hotspot</i> dan mengirimkan data ketinggian air kepada <i>Database</i> dan <i>Web</i>
2	Sensor Ultrasonik	Membaca ketinggian air sesuai parameter dan menyalurkan ke NodeMCU ESP8266 Wi-Fi	Hasil pengukuran sensor ultrasonik sesuai dengan pengukuran pada <i>serial monitor</i> dan <i>database</i>
3	Relay	Menyambungkan dan memutuskan arus pada pompa air	Pompa air dapat menyala dan digunakan
4	Pompa Air	Menyalurkan air dari dan menuju tempat yang sudah disediakan	Air dapat disalurkan dengan menggunakan selang

d. Hasil Pengujian Permukaan Air Banjir

Dalam pengujian alat, penulis sudah menetapkan batas-batas yang akan digunakan sebagai penanda ketinggian air berdasarkan referensi yang sudah dibaca [9]. Batas tersebut dapat dilihat dalam Tabel 2.

Tabel 2 Batas Ketinggian Air Permukaan

Ketinggian Air dari Permukaan	Status
0 – 10 cm	Aman

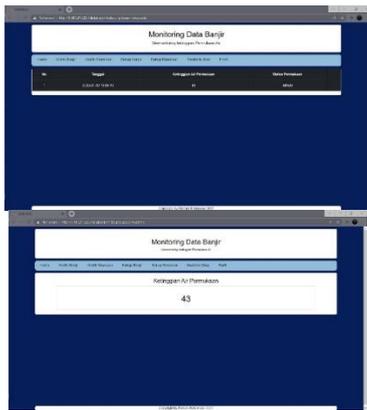
10 - 20 cm	Siaga 1
20 - 30 cm	Siaga 2
>30 cm	Banjir

Pengujian dilakukan terhadap 4 status yang sudah disiapkan. Berikut merupakan hasil yang diperoleh setelah beberapa kali percobaan.

e. Status Aman

Pada status aman, sensor ultrasonik mendeteksi ketinggian permukaan air di atas 40cm. *Serial Monitor* akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada *website*.

```
AMAN : POMPA MATI
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php
```



Gambar 6 Tampilan Data Aman: (a). Serial Monitor (b) MySQL (c). Tabel pada Website (d). Real Time pada Website

Gambar 6 merupakan tampilan dari *serial monitor* saat alat dijalankan dan akan mengirimkan data ke *database* MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke *database*(b). Kemudian tampilan pada *website*, yaitu tabel data dan *real time* data yang akan diperbarui otomatis(c,d).



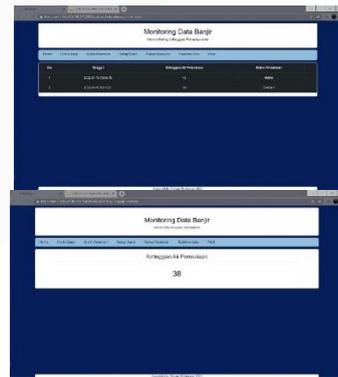
Gambar 7 Penampakan Alat saat Kondisi Aman

Gambar 7 merupakan penampakan alat saat percobaan untuk kondisi aman. Kondisi ini merupakan kondisi di mana jarak antara sensor dengan permukaan air di 0-10cm.

f. Status Siaga 1

Pada status siaga 1, sensor ultrasonik mendeteksi ketinggian permukaan air berada di antara 10-20cm. *Serial monitor* akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada *website*.

```
*****
4175: sending
DATA: 30.150000
IP Address: 192.168.42.117
Kecepatan: 100
SIAGA 1 : POMPA MATI
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php?data=30
```



Gambar 8 Tampilan Data Siaga 1: (a). Serial Monitor (b) MySQL (c). Tabel pada Website (d). Real Time pada Website

Gambar 8 merupakan tampilan dari *serial monitor* saat alat dijalankan dan akan mengirimkan data ke *database* MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil

masuk ke database(b). Kemudian tampilan pada website, yaitu tabel data dan real time data yang akan diperbarui otomatis(c,d).



Gambar 9 Penampakan Alat saat Kondisi Siaga 1

Gambar 9 merupakan penampakan alat saat percobaan untuk kondisi siaga 1. Kondisi ini merupakan kondisi di mana jarak antara sensor dengan permukaan air berada di antara 10-20cm.

g. Status Siaga 2

Pada status siaga 2, sensor ultrasonik mendeteksi ketinggian permukaan air di antara 20-30cm. Serial Monitor akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada website.

```
WiFi terhubung
SSID: Roelipnya
IP Address: 192.168.43.117
Ketinggian Permukaan :25
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php?data1=25
Ketinggian Permukaan :25
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php?data1=25
```

(a)

id	date	permukaan	status
1	2022-01-10 13:06:10	43	AMAN
2	2022-01-10 13:15:21	38	SIAGA 1
3	2022-01-10 13:20:20	25	SIAGA 2
4	2022-01-10 13:20:40	25	SIAGA 2
5	2022-01-10 13:20:55	25	SIAGA 2

(b)



(c)

(d)

Gambar 10 Tampilan Data Siaga 2: (a). Serial Monitor (b) MySQL (c). Tabel pada Website (d). Real Time pada Website

Gambar 10 merupakan tampilan dari serial monitor saat alat dijalankan dan akan mengirimkan data ke database MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke database(b). Kemudian tampilan pada website, yaitu tabel data dan real time data yang akan diperbarui otomatis(c,d).



Gambar 11 Penampakan Alat saat Kondisi Siaga 2

Gambar 11 merupakan penampakan alat saat percobaan untuk kondisi siaga 1. Kondisi ini merupakan kondisi di mana jarak antara sensor dengan permukaan air berada di antara 20-30cm.

h. Status Banjir

Pada status banjir, sensor ultrasonik mendeteksi ketinggian permukaan air di antara 30-45cm. Serial Monitor akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada website.

```
Ketinggian Permukaan :18
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php?data1=18
Ketinggian Permukaan :18
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updatepermukaan.php?data1=18
```

(a)

id	date	permukaan	status
1	2022-01-10 13:06:10	43	AMAN
2	2022-01-10 13:15:21	38	SIAGA 1
3	2022-01-10 13:20:20	25	SIAGA 2
4	2022-01-10 13:20:40	25	SIAGA 2
5	2022-01-10 13:20:55	25	SIAGA 2
6	2022-01-10 13:26:14	24	SIAGA 2
7	2022-01-10 13:26:21	18	BANJIR
8	2022-01-10 13:26:26	18	BANJIR

(b)



(c)

(d)

Gambar 12 Tampilan Data Banjir: (a). *Serial Monitor* (b) MySQL (c). Tabel pada *Website* (d). *Real Time* pada *Website*

Gambar 12 merupakan tampilan dari *serial monitor* saat alat dijalankan dan akan mengirimkan data ke *database* MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke *database*(b). Kemudian tampilan pada *website*, yaitu tabel data dan real time data yang akan diperbarui otomatis(c,d).



Gambar 13 Penampakan Alat saat Kondisi Banjir

Gambar 13 merupakan penampakan alat saat percobaan untuk kondisi banjir. Kondisi ini merupakan kondisi di mana jarak antara sensor dengan permukaan air berada di antara 30-45cm.

i. Hasil Pengujian Reservoir

Dalam pengujian alat, penulis sudah menetapkan batas-batas yang akan digunakan sebagai penanda ketinggian air dalam *reservoir*. Batas tersebut dapat dilihat dalam Tabel 3.

Tabel 3 Batas Ketinggian *Reservoir*

Ketinggian Air dari Permukaan	Status
0 – 10 cm	Aman
10 - 15 cm	Level 1
15 - 20 cm	Level 2

>20 cm	Banjir
--------	--------

Pengujian dilakukan terhadap 4 status yang sudah disiapkan. Berikut merupakan hasil yang diperoleh setelah beberapa kali percobaan.

4.1.1.1 Status Aman

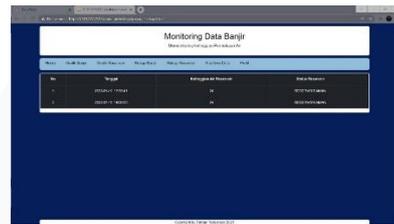
Pada status aman, sensor ultrasonik mendeteksi ketinggian permukaan air di 0-10cm. *Serial Monitor* akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada *website*.

```
Ketinggian Permukaan :24
AMAN : POMPA MATI
requesting URL: http://s.181.217.237/databanjir/db/updatereservoir.php?data2=24
Ketinggian Permukaan :24
AMAN : POMPA MATI
requesting URL: http://s.181.217.237/databanjir/db/updatereservoir.php?data2=24
```

(a)

id	date	reservoir	status
1	2022-01-10 13:59:41	24	RESERVOIR AMAN
2	2022-01-10 14:00:00	24	RESERVOIR AMAN

(b)



(c)

Gambar 14 Tampilan Data *Reservoir* Aman: (a). *Serial Monitor* (b) MySQL (c). Tabel pada *Website*

Gambar 14 merupakan tampilan dari *serial monitor* saat alat dijalankan dan akan mengirimkan data ke *database* MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke *database*(b). Kemudian tampilan pada *website*, yaitu tabel data dan *real time* data yang akan diperbarui otomatis(c,d).

j. Status Level 1

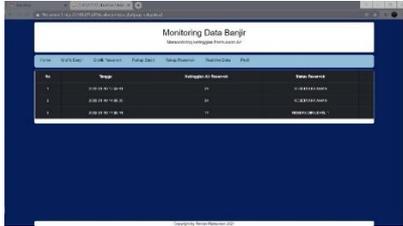
Pada status level 1, sensor ultrasonik mendeteksi ketinggian permukaan air di antara 10-15cm. *Serial Monitor* akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada *website*.

```
WiFi terhubung
SSID: Bocilpunya
IP Address: 192.168.43.63
Ketinggian RESERVOIR :17
SIAGA 1 : POMPA MATI
requesting URL: http://s.181.217.237/databanjir/db/updatereservoir.php?data2=17
```

(a)

+ Options					
	id	date	reservoir	status	
<input type="checkbox"/>	1	2022-01-10 13:59:41	24	RESERVOIR AMAN	
<input type="checkbox"/>	2	2022-01-10 14:00:00	24	RESERVOIR AMAN	
<input type="checkbox"/>	3	2022-01-10 14:05:44	17	RESERVOIR LEVEL 1	

(b)



(c)

Gambar 15 Tampilan Data Reservoir Level 1: (a). Serial Monitor (b) MySQL (c). Tabel pada Website

Gambar 15 merupakan tampilan dari serial monitor saat alat dijalankan dan akan mengirimkan data ke database MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke database(b). Kemudian tampilan pada website, yaitu tabel data dan real time data yang akan diperbarui otomatis(c,d).

k. Status Level 2

Pada status level 2, sensor ultrasonik mendeteksi ketinggian permukaan air di antara 15-20cm. Serial Monitor akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada website.

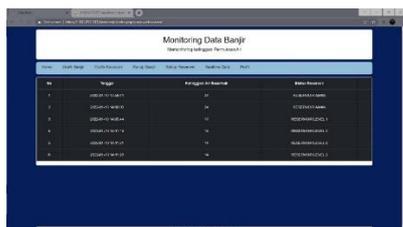
```

Ketinggian RESERVOIR :14
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=14
Ketinggian RESERVOIR :14
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=14
Ketinggian RESERVOIR :14
SIAGA 2 : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=14
    
```

(a)

+ Options					
	id	date	reservoir	status	
<input type="checkbox"/>	1	2022-01-10 13:59:41	24	RESERVOIR AMAN	
<input type="checkbox"/>	2	2022-01-10 14:00:00	24	RESERVOIR AMAN	
<input type="checkbox"/>	3	2022-01-10 14:05:44	17	RESERVOIR LEVEL 1	
<input type="checkbox"/>	4	2022-01-10 14:11:14	14	RESERVOIR LEVEL 2	
<input type="checkbox"/>	5	2022-01-10 14:11:21	14	RESERVOIR LEVEL 2	

(b)



(c)

Gambar 16 Tampilan Data Reservoir Level 2: (a). Serial Monitor (b) MySQL (c). Tabel pada Website

Gambar 16 merupakan tampilan dari serial monitor saat alat dijalankan dan akan mengirimkan data ke database MySQL. Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke database. Kemudian tampilan pada website, yaitu tabel data dan real time data yang akan diperbarui otomatis.

l Status Penuh

Pada status penuh, sensor ultrasonik mendeteksi ketinggian permukaan air di antara 20-25cm. Serial Monitor akan terkoneksi dengan WiFi yang sudah disediakan lalu mengirim data pada MySQL untuk selanjutnya diteruskan pada website.

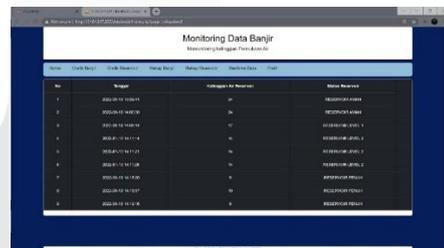
```

WiFi Koneksi
SSID: hmc1pnyaya
IP: Address: 192.168.43.63
Ketinggian RESERVOIR :10
RAJUKU : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=10
Ketinggian RESERVOIR :9
RAJUKU : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=9
Ketinggian RESERVOIR :9
RAJUKU : POMPA NYALA
requesting URL: http://5.181.217.237/databanjir/db/updateseservoir.php?data2=9
    
```

(a)

+ Options					
	id	date	reservoir	status	
<input type="checkbox"/>	1	2022-01-10 13:59:41	24	RESERVOIR AMAN	
<input type="checkbox"/>	2	2022-01-10 14:00:00	24	RESERVOIR AMAN	
<input type="checkbox"/>	3	2022-01-10 14:05:44	17	RESERVOIR LEVEL 1	
<input type="checkbox"/>	4	2022-01-10 14:11:14	14	RESERVOIR LEVEL 2	
<input type="checkbox"/>	5	2022-01-10 14:11:21	14	RESERVOIR LEVEL 2	
<input type="checkbox"/>	6	2022-01-10 14:11:28	14	RESERVOIR LEVEL 2	
<input type="checkbox"/>	7	2022-01-10 14:18:20	9	RESERVOIR PENUH	
<input type="checkbox"/>	8	2022-01-10 14:18:57	10	RESERVOIR PENUH	
<input type="checkbox"/>	9	2022-01-10 14:19:16	9	RESERVOIR PENUH	

(b)



(c)

Gambar 17 Tampilan Data Reservoir Penuh: (a). Serial Monitor (b) MySQL (c). Tabel pada Website

Gambar 17 merupakan tampilan dari serial monitor saat alat dijalankan dan akan mengirimkan data ke database MySQL(a). Lalu tampilan pada MySQL saat data hasil bacaan sensor sudah berhasil masuk ke database(b). Kemudian tampilan pada website, yaitu tabel data dan real time data yang akan diperbarui otomatis(c,d).

m. Pengujian Parameter Pada Sistem

Setelah berhasil dilakukan pengujian pada alat, kini akan dilakukan pengukuran QOS menggunakan Wireshark dan Apache JMeter. Untuk menghitung nilai Delay, Troughput, dan Packet loss pada

wireshark berdasarkan data yang diambil oleh *packet capture* menggunakan persamaan.

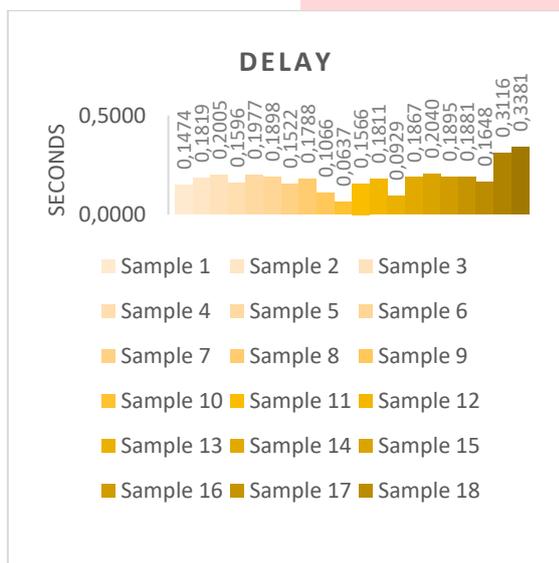
n. *Wireshark*

o. *Delay*

Berdasarkan data yang didapat dari *wireshark*, nilai *delay* dapat dihitung dengan persamaan:

$$Delay (Average) = \frac{Total\ delay\ (s)}{packets} \tag{1}$$

Penghitungan *delay* dapat dilakukan menggunakan *excel*, karena hanya memfilter bagian TCP. Gambar 18 merupakan grafik *delay* yang terjadi saat pagi, siang, dan malam. Pengujian dilakukan dengan 20 sampel.



Gambar 18 Grafik *Delay* Sistem

Berdasarkan grafik di atas, nilai *delay* terlihat naik secara konstan. Dengan rata-rata akhir *delay* sebesar 0.17959 s.

p. *Packet loss*

Berdasarkan data yang didapat dari *wireshark*, nilai *packet loss* dapat dihitung dengan persamaan

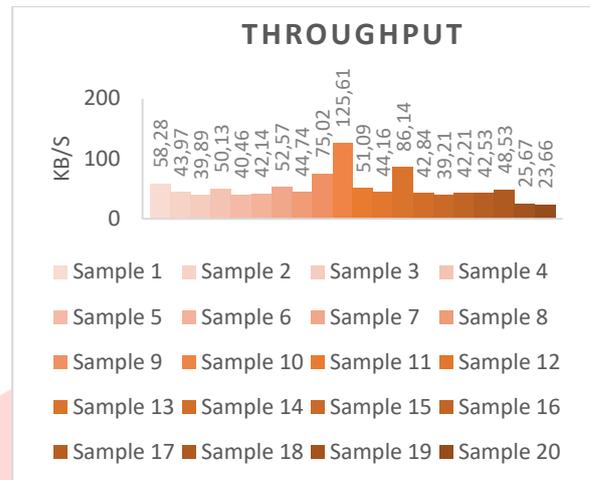
$$Packet\ Loss = \frac{Packet\ transmitted - Packet\ recieved}{Packet\ Transmitted} \times 100\% \tag{2}$$

Dengan melihat dari data statistik *loss* pada *Wireshark*, nilai rata-rata *packet loss* yang terjadi adalah 0%

q. *Throughput*

Berdasarkan data yang didapat dari *wireshark*, nilai *throughput* dapat dihitung dengan persamaan

$$Throughput = \frac{\sum packets}{Time\ Span} \tag{3}$$



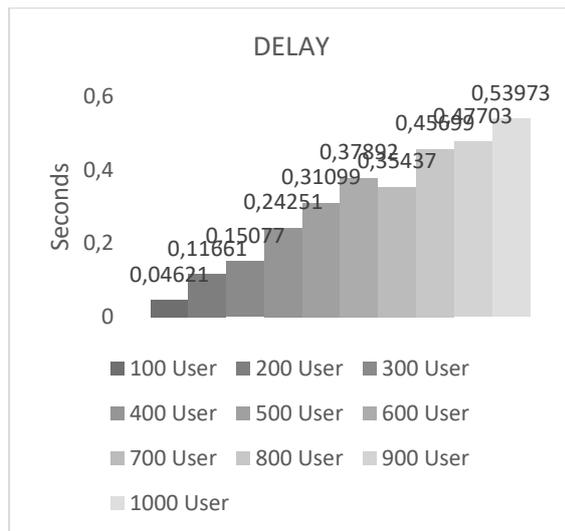
Gambar 19 Grafik *Throughput* Sistem

Gambar 19 merupakan grafik *throughput* yang terjadi selama pengujian. Sehingga rata-rata *throughput* yang terjadi dalam sistem adalah 50.94 Kb/s.

r. Apache JMeter

s. *Delay*

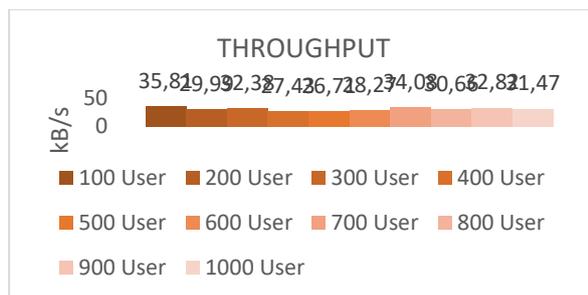
Pengujian ini dilakukan untuk melihat kemampuan VPS sebagai *server* dilakukan pengujian sebanyak 10 kali dengan melakukan perumpamaan *user* mengakses *website*. Banyak user dari 100 sampai 1000 user dengan kelipatan penambahan 100 user dan interval 100 detik setiap pengambilan data. Pada Gambar 20, grafik menunjukkan perubahan yang konstan naik dengan data pertama dalam kondisi 100 user mengakses *website* dengan *delay* sebesar 0,04621s sampai dengan pengambilan data ke-10, yaitu 0,53973s. Terdapat penurunan grafik pada pengambilan data ke-7 dengan kondisi 700 *user* sebesar 0,35437s. Dari data yang sudah didapat, rata-rata nilai *delay* sebesar 0,307413 s.



Gambar 20 Grafik Delay Apache JMeter

t. Throughput

Throughput yang didapat saat pengambilan data cenderung tidak terjadi perubahan yang sangat signifikan yang dapat dilihat pada Gambar 21. Dari data yang sudah didapat, rata-rata nilai throughput sebesar 30,96 Kb/s.



Gambar 21 Grafik Throughput Apache JMeter

V. KESIMPULAN

Penelitian ini telah mengusulkan pembuatan alat dan website terkait penanganan bencana banjir. Berdasarkan pengujian dan analisis yang sudah dilakukan, dapat disimpulkan bahwa website dan alat yang dirakit dapat berjalan sesuai dengan fungsinya. Pada pengujian sistem dengan *wreshark*, diperoleh rata-rata delay sebesar 0.17959 s, *packet loss* sebesar 0%, serta throughput sebesar 50.94 Kb/s. Pada pengujian website dengan *Apache JMeter*, diperoleh rata-rata delay sebesar 0.307413s dan throughput sebesar 30.96 Kb/s.

REFERENSI

- [1] D. Satria, S. Yana, R. Munadi, and S. Syahreza, "Peringatan Dini Banjir Secara Real-Time Berbasis Web Menggunakan Arduino dan Ethernet," *Jurnal JTIK*, vol. 1, no. 1, 2017. [Online]. Available: <http://journal.lembagakita.org/index.php/jtik>.
- [2] "Upaya Pemerintah Tanggulangi Banjir Jakarta dan Kendala yang Dihadapi Halaman all -

Kompas.com."

<https://megapolitan.kompas.com/read/2018/02/12/09432131/upaya-pemerintah-tanggulangi-banjir-jakarta-dan-kendala-yang-dihadapi?page=all> (accessed Dec. 12, 2020).

[3] M. Sholihul Hadi, D. Alfian Tricahyo, D. Kurniawan Sandy, F. Satrio Wibowo, and J. Teknik Elektro, "Iot Cloud Data Logger Untuk Sistem Pendeteksi Dini Bencana Banjir Pada Pemukiman Penduduk Terintegrasi Media Sosial," *Jurnal Edukasi Elektro*, vol. 1, no. 2, 2017, [Online]. Available: <http://journal.uny.ac.id/index.php/jee/>.

[4] I. Zuly Budiarmo, M. Cs, and E. Nurraharjo, "Sistem Monitoring Tingkat Ketinggian Air Bendungan Bebas Mikrokontroler," *J. Din. Inform*, vol. 3, no. 1, 2012.

[5] R. Fikri, B. P. Lapanoro, D. Muh, and I. Jumarang, "Rancang Bangun Sistem Monitoring Ketinggian Permukaan Air Menggunakan Mikrokontroler ATMEGA328P Berbasis Web Service," *Positron*, vol. V, no. 2, pp. 42–49, 2015.

[6] I. Fitri Astuti, A. N. Manoppo, Z. Arifin, and I. Komputer, "Sistem Peringatan Dini Bahaya Banjir Kota Samarinda Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler Dengan Buzzer Dan Sms," *Sebatik Stmik Wicida*, vol. 22, pp. 30–34, 2018.

[7] Loizou, K., Koutroulis, E., Zalikas, D., & Liontas, G. (2015). A Low-cost Capacitive Sensor for Water Level Monitoring in Large-Scale Storage Tanks, 2015 ICIT. <https://doi.org/10.1109/ICIT.2015.7125295>

[8] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018, doi: 10.1109/TII.2018.2852491.

[9] Indianto, W., & Harsa Kridalaksana, A. (2017). Perancangan Sistem Prototipe Pendeteksi Banjir Peringatan Dini Menggunakan Arduino Dan PHP. *Jurnal Informatika Mulawarman*, 12(1), 45–49.