

Monitoring Ketinggian Air dan Curah Hujan Dalam *Early Warning System* Bencana Banjir Berbasis IoT

Ivander Achmad Wandi*¹, Ahmad Ashari*²

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹ivanderiaw@mail.ugm.ac.id, ²ashari@ugm.ac.id

Abstrak

Banjir merupakan bencana alam dimana terdapat volume air yang berlebihan sehingga merendam daratan. Jika dibiarkan banjir dapat membawa penyakit, menyebabkan kerusakan lingkungan serta menghambat mobilitas masyarakat. Oleh karena itu, dibutuhkan suatu sistem yang dapat memberikan peringatan dini kepada masyarakat sebelum terjadi banjir. Seiring perkembangan teknologi, kita dapat memantau ketinggian air di suatu daerah untuk mengantisipasi terjadinya banjir. Konsep ini lah yang disebut *Early Warning System (EWS)*. IoT dapat membantu pemantauan dan peringatan banjir secara langsung dan terus menerus. IoT juga dapat memantau ketinggian air dari jarak jauh. Penelitian protokol MQTT serta menggunakan sensor ketinggian air dan curah hujan sebagai data untuk deteksi banjir. Sistem ini akan mendeteksi ketinggian air dan curah hujan secara real time. Apabila ketinggian air dan curah hujan mencapai batas tertentu maka sistem akan memberikan peringatan kepada warga sekitar melalui chatbot. Terdapat 3 jenis peringatan pada sistem ini. Hasil pemantauan juga ditampilkan melalui dashboard dan hasil bacaan sensor akan disimpan di dalam database cloud. Hasil evaluasi menunjukkan bahwa sistem yang dirancang dapat bekerja dengan baik dalam memberikan peringatan dini banjir. Bot pada telegram juga sudah mengirimkan notifikasi dengan delay rata rata 0.561 detik.

Kata kunci— IoT, MQTT, EWS, Banjir

Abstract

Floods are natural disasters where there is an excessive volume of water that dampens the land. If left unchecked, flooding can bring disease, cause environmental damage and hinder people's mobility. Therefore, we need a system that can provide early warning to the community before a flood occurs. As technology develops, we can monitor the water level in an area to anticipate flooding. This concept is called the *Early Warning System (EWS)*. IoT can help with real-time and continuous monitoring and warning of floods. IoT can also monitor air levels remotely. The MQTT protocol research also uses water level and rainfall sensors as data for flood detection. This system will detect water level and rainfall in real time. If the water level and rainfall reach a certain limit, the system will provide a warning to local residents via chatbot. There are 3 types of alerts in this system. Monitoring results are also displayed via the dashboard and sensor readings will be stored in the cloud database. The evaluation results show that the designed system can work well in providing flood early warning. Bots on Telegram have also sent notifications with an average delay of 0.561 seconds.

Keywords— IoT, MQTT, EWS, Flood

1. PENDAHULUAN

Banjir merupakan persoalan yang cukup sering didengar. Persoalan ini terdengar biasa tetapi masih sering terjadi. Banjir biasanya terjadi karena curah hujan yang tinggi atau sistem irigasi yang kurang baik. Banjir sendiri membawa banyak masalah seperti penyakit, kerusakan lahan dan bangunan dan menghambat mobilitas masyarakat. Melihat dampak buruk yang ditimbulkan oleh banjir. Diperlukan sistem yang dapat memantau secara continuous untuk menghasilkan data yang lebih baik dan metode untuk membantu analisis secara real time dalam memprediksi banjir [1]. Disini lah peran Internet of Things (IoT) dibutuhkan. IoT merupakan suatu sistem yang terdiri dari beberapa perangkat yang saling terhubung dan terhubung ke internet. IoT banyak digunakan pada sistem yang memerlukan pengambilan dan pemantauan data secara real time. Pada deteksi awal banjir, IoT untuk membantu dalam mengembangkan sistem untuk mengukur dan mengirim informasi dan notifikasi kepada pengguna [2].

Berdasarkan penelitian sebelumnya, dalam deteksi awal banjir, diperlukan sensor yang dapat memantau ketinggian air. Pemantauan ketinggian air dapat dilakukan menggunakan sensor HC-SR04. Sensor ini bekerja menggunakan gelombang suara yang dipantulkan dari suatu objek ke objek lain dan dipantulkan kembali ke sensor ultrasonic. Sensor ini memiliki akurasi yang baik dengan tingkat kesalahan sekitar 1 cm [3] Sehingga cocok digunakan untuk pemantauan ketinggian air. Selain menggunakan sensor tersebut Sensor level juga digunakan dalam monitoring ketinggian air.

Salah satu faktor utama dari kenaikan ketinggian air adalah hujan. Hujan terus menerus dan sistem irigasi air yang buruk akan mempercepat terjadinya banjir. Oleh karena itu diperlukan sensor curah hujan yang akan mengambil data curah hujan [4].

Selain melakukan deteksi awal terhadap banjir. Sistem perlu mengirim notifikasi kepada pengguna untuk bersiap siap menghadapi banjir. Selain itu, Aplikasi yang digunakan tentunya harus banyak digunakan oleh masyarakat. Berdasarkan penelitian [5], Sistem notifikasi yang dibuat akan memberikan peringatan kepada warga yang berada di sekitar lokasi yang terdeteksi banjir. Hal ini sudah dilakukan oleh [6].

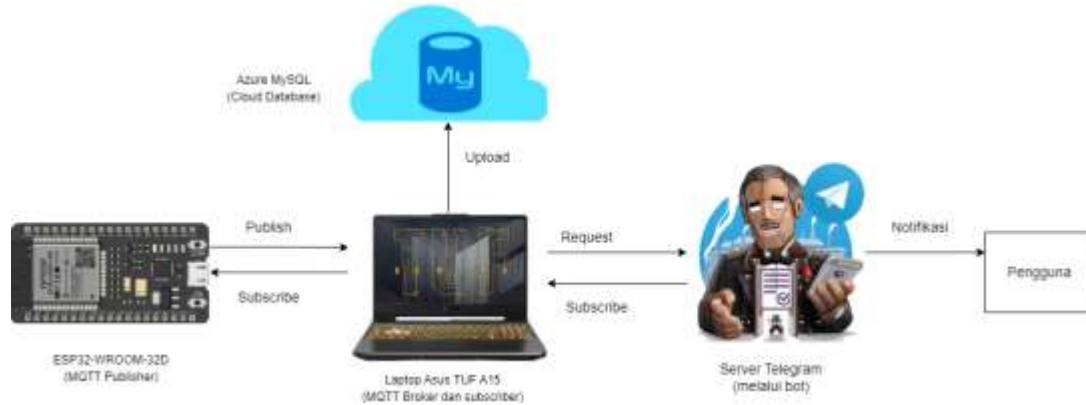
2. METODE PENELITIAN

2.1 Analisa Sistem

Sistem ini bertujuan memberikan peringatan dini terhadap bencana banjir. Peringatan ini disertai dengan lokasi tempat yang berpotensi terjadinya banjir Bencana banjir dapat diketahui jika ketinggian air meningkat secara cepat dan curah hujan yang tinggi. Pada penelitian ini, ESP32 WROOM 32D berperan sebagai publisher yang akan mengakuisisi data dari sensor, mendeteksi banjir lalu mengirim hasilnya ke laptop. Laptop berperan sebagai broker dan subscriber yang akan mengolah data lebih lanjut dan mengirim notifikasi melalui telegram. Notifikasi yang dikirim berupa peringatan dan lokasi terjadinya banjir. Data yang sudah diolah akan dikirim ke cloud.

2.2 Rancangan Model dan Hardware Sistem

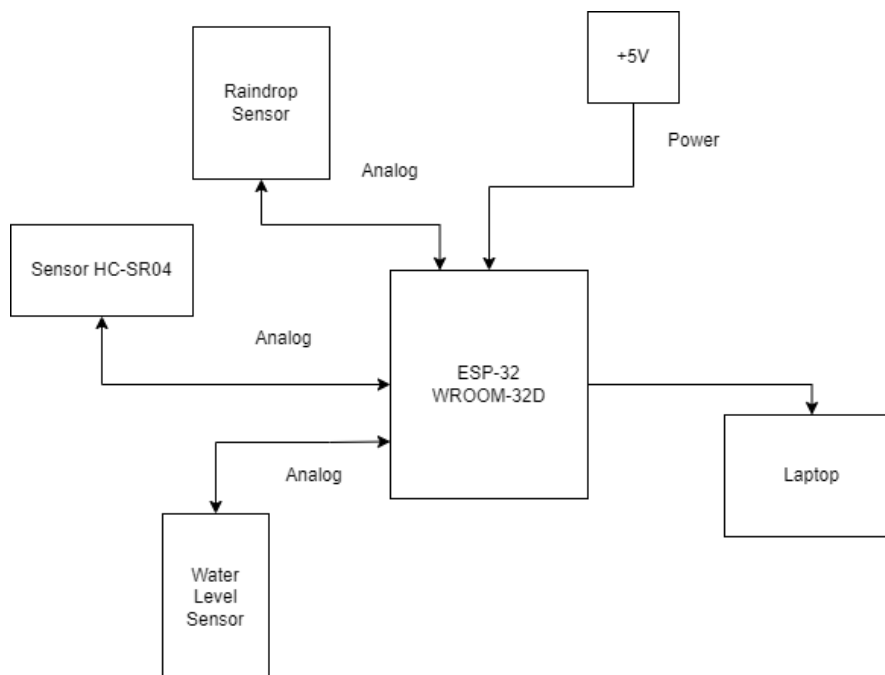
Rancangan model sistem internet of things secara garis besar dapat dilihat pada Gambar 1 ESP32 WROOM akan menerima data untuk selanjutnya di-publish ke laptop. Kemudian Laptop akan melakukan komputasi terhadap data yang diterima. Jika hasil pengolahan data mendeteksi kemungkinan terjadinya banjir maka laptop akan mengirimkan notifikasi menggunakan bot telegram. Data yang sudah diolah akan dikirimkan ke cloud.



Gambar 1 Diagram Sistem IoT

Untuk Akuisisi data dilakukan oleh ESP32 WROOM. Mikrokontroler ini dipilih karena Dibandingkan dengan ESP8266, ESP32-WROOM-32 memiliki lebih banyak pin analog. Selain itu mikrokontroler ini memiliki antenna yang tertanam untuk memindai seluruh jaringan wifi disekitarnya. Hal ini membuat ESP32 WROOM-32 cocok untuk berbagai platform dan aplikasi IoT [7]. Untuk pengukuran ketinggian air digunakan sensor HC-SR04. Sensor ini memiliki range pengukuran 2 cm sampai 400 cm. Pada percobaan yang dilakukan [8], Sensor HC-SR04 memiliki rata rata error di 1.14% dan nilai akurasi adalah 98.86%. Untuk Sensor curah hujan yang digunakan adalah Sensor MH-RD an untuk level sensor yang digunakan adalah level sensor kapasitif.

Rancangan akuisisi data pada ESP32 WROOM dapat dilihat pada Gambar 2. Sensor yang digunakan adalah HC-SR04, sensor level dan sensor curah hujan. HC-SR04 dan sensor level akan mengirim data berupa ketinggian air saat ini. Digunakan 2 sensor untuk hasil pengukuran yang lebih baik. Sensor level akan diletakan pada ketinggian 21 cm dari dasar wadah. Sensor curah hujan akan mengirim data berupa curah hujan dalam persen. ESP 32 WROOM akan membuat topic pada tiap hasil bacaan untuk dapat di subscribe.



Gambar 2 Diagram Akuisisi data Pada ESP32 WROOM-32D

2.3 Rancangan Perangkat Lunak Sistem

2.3.1 MQTT

Message Queue Telemetry Transport (MQTT) merupakan protokol yang digunakan pada komunikasi lingkungan IoT yang bekerja pada *Transport layer*. MQTT menggunakan model komunikasi *publish-subscribe*, dimana klien tidak memerlukan update, sehingga mengurangi penggunaan *resource*. Subscribe adalah proses dimana client berlangganan pesan dari broker. Publish adalah proses saat client mempublikasikan pesan ke broker. Publisher akan di program menggunakan arduino IDE dengan bahasa C dengan library Pub/Sub Client. Sebelum mengirim data, publisher sudah harus terhubung ke broker melalui alamat IP. Selanjutnya program akan membuat topik untuk setiap. keluaran sensor yang diterima ESP32 WROOM.

Broker dan subscriber pada sistem ini adalah laptop yang di program dengan Node RED. Node-RED akan mengatur data yang dikirim publisher, mengirim data ke cloud, membuat dashboard dan mengirim notifikasi melalui telegram. Laptop akan menerima data dari publisher dan mengolahnya berdasarkan threshold dari ESP32. Data yang didapat dari publisher akan dimasukan kedalam variable dengan nama yang sama dengan topiknya. Pengukuran utama dari sistem ini adalah sensor HC-SR04 dan sensor curah hujan.

Ada 3 jenis alert atau notifikasi pada sistem ini, seperti yang terlihat pada Tabel 1. Alert yang pertama berisi ketinggian air dan curah hujan saat ini. Alert kedua berisi ketinggian air, curah hujan dan lokasi yang berpotensi banjir. Alert ketiga berisi ketinggian air, curah hujan dan lokasi yang berpotensi banjir. Pada alert ketiga, user akan diminta bersiap-siap menghadapi banjir. Pada paper [9] digunakan nilai threshold 14 cm dan 18 cm. Output algoritma tersebut dapat dilihat pada Tabel 1

Tabel 1 Output Algoritma Deteksi Banjir

level1/ rain	<35%	>=35%
<14cm	No alert	Alert 1
>=14cm, <=18	Alert 1	Alert 2
>18	Alert 2	Alert 3

2.3.2 Bot Telegram

Pada sistem ini. Telegram digunakan sebagai media komunikasi untuk mengirim data ke pengguna. Telegram dipilih karena telegram adalah aplikasi layanan pengirim pesan dengan fokus pada kecepatan dan keamanan. Telegram dapat mengirim pesan, foto, video dan file jenis jenis apapun. Telegram juga mendukung panggilan end-to-end terenkripsi [10] Untuk membuat bot pada telegram, pertama ketik “newbot” pada chat dengan botfather. Selanjutnya bot father akan meminta nama bot dan username nya. Setelah itu, botfather akan memberikan id dari bot. Simpan id dari bot yang sudah dibuat. Bot bisa digunakan dengan cara memakai chatid yang sudah didapat pada Node-RED.

2.3.3 Azure Cloud

Sistem IoT ini menggunakan cloud dalam penyimpanan data. Data yang dikirim ke cloud adalah ketiga berisi ketinggian air, curah hujan, latitude, longitude dan tipe alert yang dikirim (0-3). Sebelum data dapat dikirim, pertama tama harus dibuat suatu SQL Database yang akan menjadi tempat penyimpanan data. Selanjutnya gunakan IP address VM tadi untuk komunikasi dengan broker melalui Node-RED.

2.4 Kalibrasi Sensor

Terdapat 3 sensor yang akan dikalibrasi, kalibrasi dilakukan untuk mengukur tingkat keakuratan dan kelayakan sensor. Pada sensor HC-SR04, penggunaan library NewPing.h sudah

dan median filter sudah membuat sensor cukup akurat. Nilai analog pada sensor curah hujan MH-RD memiliki nilai awal 4095 dan akan terus menurun. Nilai ini cukup fluktuatif sehingga ditambahkan filter *Exponential Weighted Moving Average (EWMA)* untuk menstabilkan nilai analog yang didapat. Setelah kalibrasi dilakukan, diketahui bahwa sensor curah hujan ini memiliki rentang nilai analog 4095 – 1880. Untuk mengubahnya menjadi persen menggunakan persamaan (1)

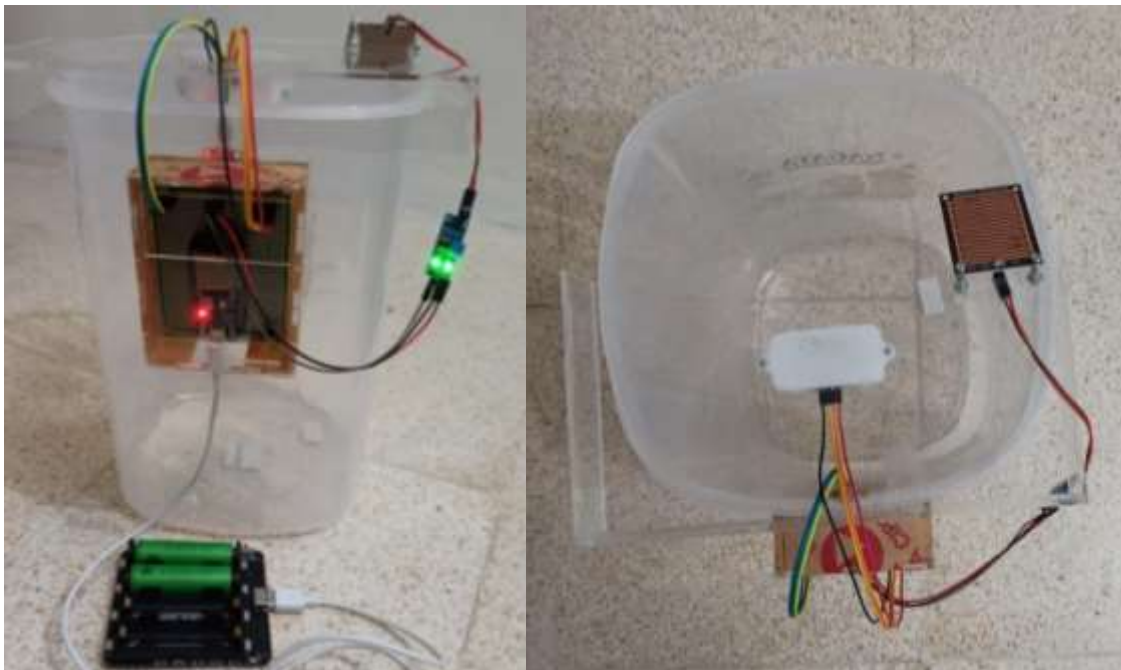
$$rain = \frac{4095 - nilai_{analog}}{2115} \times 100 \quad (1)$$

Sensor level memiliki nilai analog awal 0 dan nilai ini akan terus bertambah sesuai dengan ketinggian air. Setelah kalibrasi dilakukan, diketahui bahwa sensor level ini memiliki rentang nilai analog 0 – 2160

$$level2 = \frac{nilai_{analog} \times 100}{2160} \quad (2)$$

2.5 Akusisi data pada ESP32

Pada penelitian ini, akusisi data dilakukan pada wadah plastik ukuran Semua perangkat keras akan dimasukan ke wadah plastik ukuran 21 × 22 × 30 cm yang ditutup sebagian dengan akrilik. Sensor dan ESP32 diletakan pada suatu papan PCB. ESP32 akan menyediakan daya untuk semua sensor. ESP32 sendiri akan diberi daya oleh battery 18650 melalui kabel USB. Untuk sensor HC-SR04 sudah cukup dengan library NewPing.h dan median filter. Sensor curah hujan menggunakan filter *Exponential Weighted Moving Average (EWMA)* untuk menstabilkan data Sementara sensor level menggunakan *raw* data saja. Papan pcb tadi diltekan pada box akrilik, sensor curah hujan dan sensor HC-Sr04 dilteakan di tutup akrilik pada bagian atas wadah plastik. Tampilan sistem secara keseluruhan dapat dilihat pada Gambar 3.



Gambar 3 Sistem Secara Keseluruhan

2.6 Konfigurasi Broker

Broker bertugas menerima data, mengirim notifikasi, menghitung delay notifikasi, mengupload ke cloud dan menampilkan dashboard. Broker menerima data menggunakan mosquitto 3.1.1. Dengan bantuan node-red, broker akan men-*subscribe* topik dari publisher. Setiap topik tadi disimpan di *flow* variable untuk selanjutnya diolah. Jika topik *alert* bernilai 0 maka node-RED tidak akan mengirimkan notifikasi tetapi jika node-red mendapatkan alert 1-3, akan dikirim notifikasi melalui telegram. Selanjutnya akan dihitung delay pengiriman notifikasi telegram. Perhitungan dilakukan dengan mengurangi *timestamp* saat data dikirim dan *timestamp* saat data diterima. Data yang didapat akan di upload ke cloud.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Deteksi Banjir dan Delay Telegram

Pertama dilakukan pengujian perubahan ketinggian air. Bagian atas wadah plastik yang terbuka akan menjadi tempat masuknya selang air. Selanjutnya kran dinyalakan dan air akan mengalir sehingga ketinggian air naik. Perubahan ketinggian air ini dikombinasikan dengan 5 nilai dari sensor curah hujan. Selanjutnya publisher akan mengirim data ke broker dan akan ditampilkan pada dashboard. Hasil percobaan deteksi banjir dan delay telegram dapat dilihat pada Tabel 2, Tabel 3, Tabel 4, Tabel 5, dan Tabel 6.

Tabel 2 Deteksi banjir saat curah hujan 0%

No	Ketinggian air (cm)	Output Sensor Level (%)	Tipe Alert (0-3)	Delay telegram (s)
1	0	0	0	0
2	3	0	0	0
3	6	0	0	0
4	9	0	0	0
5	12	0	0	0
6	15	0	1	0.456
7	18	0	1	0.493
8	21	17	2	0.437
9	24	86	3	0.462

Tabel 3 Deteksi banjir saat curah hujan 25%

No	Ketinggian air (cm)	Output Sensor Level (%)	Tipe Alert (0-3)	Delay telegram (ms)
1	0	0	0	0
2	3	0	0	0
3	6	0	0	0
4	9	0	0	0
5	12	0	0	0
6	15	0	1	0.422
7	18	0	1	0.452
8	21	0	2	0.458
9	24	83	3	1.149

Tabel 4 Deteksi banjir saat curah hujan 45%

No	Ketinggian air (cm)	Output Sensor Level (%)	Tipe Alert (0-3)	Delay telegram (ms)
1	0	0	1	0.641
2	3	0	1	0.471
3	6	0	1	0.523
4	9	0	1	0.693
5	12	0	1	0.456
6	15	0	1	0.746
7	18	0	2	0.496
8	21	22	3	0.557
9	24	81	3	0.502

Tabel 5 Deteksi banjir saat curah hujan 70%

No	Ketinggian air (cm)	Output Sensor Level (%)	Tipe Alert (0-3)	Delay telegram (ms)
1	0	0	1	0.468
2	3	0	1	0.548
3	6	0	1	0.517
4	9	0	1	0.481
5	12	0	1	0.665
6	15	0	1	1.025
7	18	0	1	0.54
8	21	33	2	0.73
9	24	71	3	0.561

Tabel 6 Deteksi banjir saat curah hujan 90%

No	Ketinggian air	Output Sensor Level	Tipe Alert	Delay telegram
1	0	0	0	0.46
2	3	0	0	0.51
3	6	0	0	0.628
4	9	0	0	0.481
5	12	0	0	0.506
6	15	0	1	0.506
7	18	0	1	0.613
8	21	52	2	0.6075
9	24	94	3	0.4815

3.2 Database di Cloud

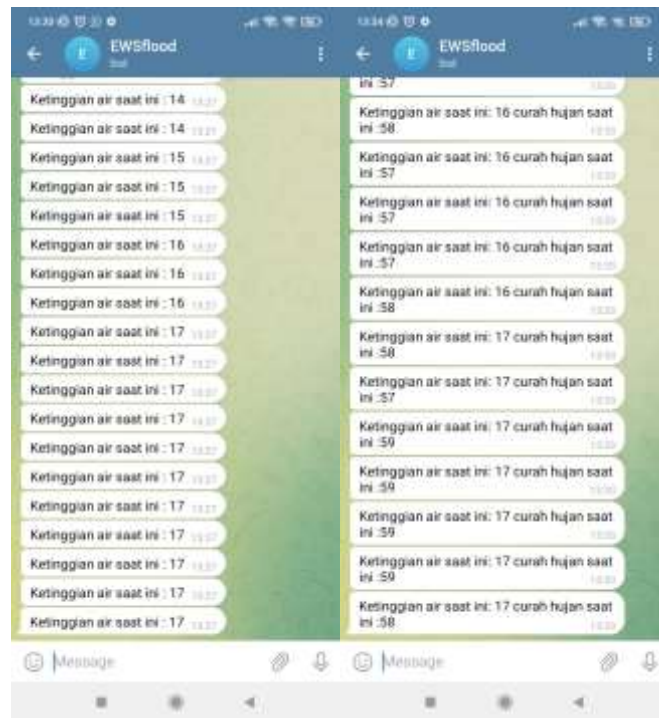
Untuk melihat hasil penelitian di *cloud*, jalankan query pada Gambar 4 di menu query editor di di cloud. Untuk mengunduh hasil query tersebut, klik “export” lalu pilih “to csv”



Gambar 4 Data yang di Upload Cloud

3.3 Pengujian Notifikasi Telegram

Hasil notifikasi yang dikirimkan ke telegram ditunjukkan pada Gambar 5 dan Gambar 6.



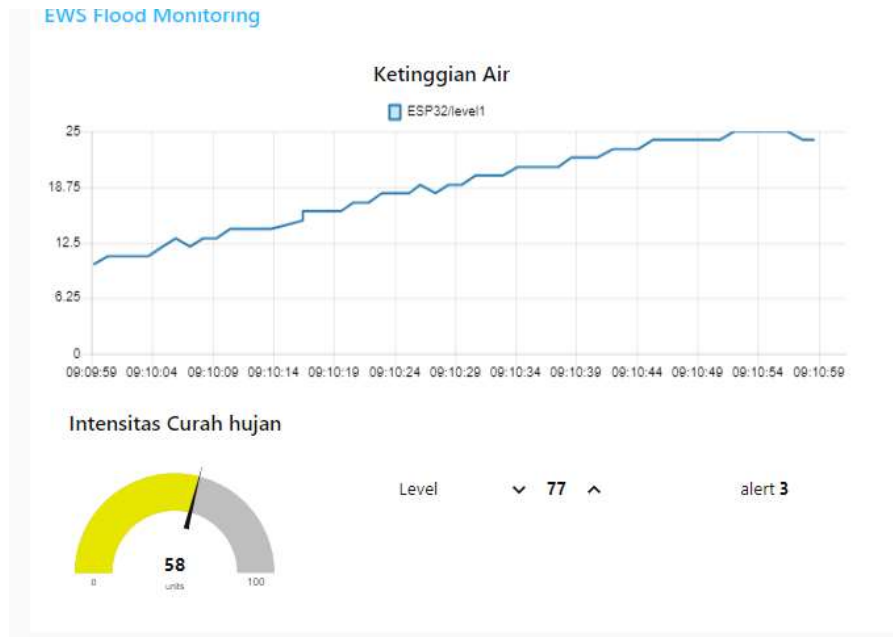
Gambar 5 Notifikasi Telegram Alert = 1 (kiri) dan Alert = 2 (kanan)



Gambar 6 hasil Notifikasi Pada Telegram Alert = 3

3.4 Tampilan Dashboard

Dashboard yang dibuat cukup responsif. Dashboard berhasil menampilkan data secara realtime. Untuk grafik ketinggian air, dashboard di atur agar menampilkan data dalam 1 menit terakhir. Tampilan dashboard dapat dilihat pada Gambar 7



Gambar 7 Tampilan Dashboard

3.5 Pembahasan

Penelitian ini menguji implementasi Early Warning Sistem (EWS) berbasis IoT dalam mendeteksi potensi banjir dan memberikan notifikasi awal melalui aplikasi telegram. Hasil penelitian menunjukkan bahwa sensor HC-SR04 sudah memberikan hasil bacaan yang akurat. Saat keran dinyalakan, sensor sudah dapat membaca ketinggian air dan perubahannya secara akurat. Sementara itu, hasil bacaan sensor curah hujan masih sedikit fluktuatif. Hal ini dipengaruhi oleh nilai analog sensor yang memiliki rentang cukup besar serta posisi meletakkan sensor yang dibuat miring sehingga ada kemungkinan air mengalir sehingga nilainya berubah ubah ditambah lagi sensor ini tidak bersifat linear seperti sensor HC-SR04. Oleh karena itu perlu ditambahkan filter EWMA agar hasil bacaan sensor lebih akurat. Untuk bacaan sensor level masih kurang akurat karena sensor ini akan menjadi cadangan jika ketinggian air mencapai lebih dari atau sama 21 cm. Secara keseluruhan, bagian akuisisi data memiliki respon yang cukup cepat terhadap perubahan ketinggian dan perubahan curah hujan.

Dari sisi broker, data akan diterima mosquitto dan diambil oleh Node-RED. Penggunaan flow dapat memudahkan proses pengolahan data dan debugging. Hal ini karena pada node node yang mengolah data diberikan node debug yang akan mengirimkan output dari setiap node. Output ini juga termasuk error yang dikeluarkan suatu node. Node ini juga digunakan pada output node azure-cloud untuk melihat error saat upload data. Dashboard yang dibuat pun cukup sederhana tetapi mudah digunakan. Pada dashboard ini dapat dilihat semua topik MQTT yang diterima oleh Node-RED. Ketinggian air dapat dilihat dalam bentuk grafik sehingga dapat dilihat perkembangannya tiap menit sementara topik rain dan level2 menggunakan tipe gauge. Akses ke database di cloud cukup mudah. Dapat melalui azure data studio atau langsung masuk ke halaman web azure cloud.

Notifikasi yang dikirim lewat telah sesuai dengan *alert* yang dideteksi. Saat *alert* bernilai 0 maka program tidak mengirim notifikasi. Saat *alert* bernilai 1 maka program mengirim notifikasi berupa ketinggian air saat ini. Saat *alert* bernilai 2 maka program mengirim notifikasi berupa ketinggian air dan curah hujan saat ini. Saat *alert* bernilai 3 maka program mengirim notifikasi berupa ketinggian air, curah hujan saat ini serta peringatan untuk mengungsi dalam cetak tebal. Pengiriman notifikasi telegram memiliki rata rata delay sebesar 0.561 detik dengan delay terkecil yaitu 0.01 detik dan delay terbesar adalah 1.224. Hasil menunjukkan bahwa delay yang didapatkan lebih baik daripada menggunakan sms seperti pada

penelitian [11]. Delay ini dapat berubah-ubah tergantung kondisi sinyal dan provider yang digunakan.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan terkait monitoring ketinggian air dan curah hujan dalam early warning sistem bencana banjir berbasis iot, dapat diambil kesimpulan sebagai berikut. Penggunaan sensor ketinggian air dan sensor curah hujan sangat diperlukan dalam pembuatan sistem *Early Warning System* untuk mendeteksi potensi banjir dengan baik. Pengiriman notifikasi melalui telegram memiliki delay rata rata 0.561 detik. Nilai ini lebih kecil dibandingkan menggunakan SMS. Penggunaan database pada cloud membantu analisa data.

DAFTAR PUSTAKA

- [1] L. Saravanan, W. Nancy, K. P. Chandran, D. Vijayanandh, J. R. Arunkumar, and R. T. Prabhu, "A Novel Approach for a Smart Early Flood Detection and Awareness System using IoT," *IEEE Xplore*, Apr. 01, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9782286>.
- [2] Y. Suppakhun, "Flood surveillance and alert system an advance the IoT," *IEEE Xplore*, Nov.01, 2019. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8953179>.
- [3] L. Efendi and W. Wildian, "Rancang Bangun Sistem Deteksi dan Informasi Lokasi Banjir Berbasis GSM," *Jurnal Fisika Unand*, vol. 7, no. 4, pp. 328–333, Oct. 2018, doi: <https://doi.org/10.25077/jfu.7.4.328-333.2018>.
- [4] S. shankar, J. J. Jijesh, D. R. Bolla, M. Penna, P. V. Sruthi, and A. Gowthami, "Early Detection of Flood Monitoring and Alerting System to Save Human Lives," 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Nov. 2020, doi: <https://doi.org/10.1109/rteict49044.2020.9315556>.
- [5] R. A. MANNA and S. GHOSH, "A COMPARATIVE STUDY BETWEEN TELEGRAM AND WHATSAPP IN RESPECT OF LIBRARY SERVICES," *INTERNATIONAL JOURNAL OF LIBRARY AND INFORMATION SCIENCE*, vol. 7, no. 2, Mar. 2018, doi: <https://doi.org/10.34218/ijlis.7.2.2018.001>.
- [6] V. Babu and V. Rajan, "Flood and Earthquake Detection and Rescue Using IoT Technology," 2019 International Conference on Communication and Electronics Systems (ICCES), Jul. 2019, doi: <https://doi.org/10.1109/icces45898.2019.9002406>.
- [7] M. Oturak and E. Dursun, "A Cost-Effective IoT Based Smart Home Application," 2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Aug. 2021, doi: <https://doi.org/10.1109/inista52262.2021.9548468>.
- [8] P. D. Widayaka, S. Hadi, R. P. M. D. Labib, and K. Marzuki, "Komparasi Performansi Sensor sebagai Perangkat Pengukuran Ketinggian Air pada Sistem Notifikasi Banjir," *Jurnal Bumigora Information Technology (BITE)*, vol. 4, no. 1, pp. 37–48, Jun. 2022, doi: <https://doi.org/10.30812/bite.v4i1.1997>.
- [9] MI. Hadi *et al.*, "Designing Early Warning Flood Detection and Monitoring System via IoT," *IOP Conference Series: Earth and Environmental Science*, vol. 479, p. 012016, Jul. 2020, doi: <https://doi.org/10.1088/1755-1315/479/1/012016>.
- [10] M. Yusuf Efendi and J. Eka Chandra, "Sistem Pemantauan dan Pengendali Lampu Ruangan Laboratorium Berbasis NodeMCUESP8266 dengan Aplikasi Telegram Bot," *RELE (Rekayasa Elektrikal dan Energi) : Jurnal Teknik Elektro*, vol. 4, no. 2, Jan. 2022, doi: <https://doi.org/10.30596/rele.v4i2.9562>.
- [11] W. Md. Shah, F. Arif, A. A. Shahrin, and A. Hassan, "The Implementation of an IoT-Based Flood Alert System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, 2018, doi: <https://doi.org/10.14569/ijacsa.2018.091187>.