

Sistem *monitoring* pada jaringan sensor banjir jalan raya menggunakan protokol MQTT

Qory Hidayati^{1*}, Nurwahidah Jamal², Fajar Aqhari Bolang³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Balikpapan

Jl. Soekarno Hatta Km. 8 Karang Joang, Balikpapan, Kalimantan Timur, Indonesia

^{1*}qory.hidayati@poltekba.ac.id, ²nurawahidah.jamal@poltekba.ac.id, ³fajaraqhari@gmail.com

ABSTRAK

Kejadian banjir di jalan raya menyebabkan masyarakat tidak dapat mengetahui apakah jalur tersebut aman untuk dilalui atau tidak dan tidak jarang terjadi kecelakaan akibat hal tersebut. Pada penelitian ini, penulis membuat sistem *monitoring* pada jaringan sensor banjir untuk jalan raya. Data dari jaringan sensor akan dikirimkan melalui modul *Long Range* (LoRa). Sistem terhubung ke jaringan internet melalui modul NodeMCU ESP8266 dan mengirimkan data sensor ke *server* menggunakan protokol *Message Queuing Telemetry Transport* (MQTT). Data ditampilkan pada *website* dalam bentuk grafik serta mengirimkan notifikasi Telegram dimana kedua sistem tersebut bekerja melalui aplikasi Node-Red. Pengujian dilakukan dengan tiga tahap yaitu konektivitas, fungsionalitas, dan *delay*. Uji konektivitas membuktikan bahwa NodeMCU ESP8266 dapat terhubung ke jaringan internet dengan jarak maksimal 30 meter dari *access point*. Uji fungsionalitas membuktikan bahwa LoRa dapat mengirimkan dan menerima paket data dari jaringan sensor sampai dengan jarak 350 meter. Pengujian *delay* menunjukkan bahwa waktu yang dibutuhkan untuk satu kali pengiriman data adalah rata-rata 0,042 detik. Sistem *monitoring* pada jaringan sensor banjir diharapkan dapat memberikan informasi ketinggian banjir dan membantu masyarakat dalam upaya meminimalisir kerugian akibat banjir.

Kata kunci: *monitoring* banjir, Node-Red, NodeMCU ESP8266, LoRa, protokol MQTT

ABSTRACT

Flood events on the highway cause people to not be able to know whether the route is safe to pass or not and accidents are not uncommon as a result. In this study, the author makes a monitoring system on a flood sensor network for highways. Data from the sensor network will be sent via the Long Range (LoRa) module. The system is connected to the internet network via the NodeMCU ESP8266 module and sends sensor data to the server using the Message Queuing Telemetry Transport (MQTT) protocol. The data is displayed on the website in graphic form and sends Telegram notifications where both systems work through the Node-Red application. The test is carried out in three stages, namely connectivity, functionality, and delay. The connectivity test proves that the NodeMCU ESP8266 can connect to the internet network with a maximum distance of 30 meters from the access point. The functionality test proves that LoRa can send and receive data packets from the sensor network up to a distance of 350 meters. The delay test shows that the time required for one time data transmission is an average of 0.042 seconds. The monitoring system on the flood sensor network is expected to be able to provide information on flood heights and help the community in an effort to minimize losses due to flooding.

Keywords: *flood monitoring, Node-Red, NodeMCU ESP8266, LoRa, MQTT protocol*

1. PENDAHULUAN

Salah satu bencana alam yang kerap terjadi di berbagai daerah Indonesia adalah banjir. Bencana banjir sering terjadi di jalan raya rawan banjir yang diakibatkan kondisi saluran drainase yang banyak tersumbat, beban berlebih, dan rendahnya mutu konstruksi [1]. Banjir di jalan raya menyebabkan masyarakat tidak dapat mengetahui apakah jalur tersebut aman untuk dilalui atau tidak dan tidak jarang terjadi kecelakaan akibat hal tersebut. Permasalahan ini nampak tidak ada pencegahan secara efektif untuk meminimalisir korban jiwa, serta masih minimnya sistem untuk memberi informasi kondisi banjir secara *online* dan *real-time*.

Beberapa sistem dan teknologi yang berkaitan dengan banjir telah dirancang dalam beberapa penelitian sebelumnya. Sistem peringatan dini banjir dengan memantau ketinggian air sungai dari jarak jauh dapat dilakukan melalui aplikasi berbasis *web* [2]. Sistem tersebut memanfaatkan

mikrokontroler Raspberry Pi untuk mengambil data ketinggian air sungai dari penangkapan sensor ultrasonik HC-SR04 dan akan dikirimkan ke *database* menggunakan protokol *Message Queuing Telemetry Transport* (MQTT). Penelitian yang serupa juga dilakukan dengan membuat sistem pendeteksi dini banjir yang memanfaatkan sensor ultrasonik untuk memantau ketinggian air sungai dan juga memanfaatkan jaringan Wi-Fi untuk mengirimkan informasi ketinggian air sungai dengan menggunakan komponen modul ESP8266. Sistem ini menggunakan aplikasi Blynk sebagai media *monitoring* dan Twitter sebagai media notifikasi [3]. Kekurangan dari sistem sebelumnya adalah pemantauan hanya dilakukan pada ketinggian sungai atau banjir tanpa memberikan informasi apakah jalan raya yang rawan banjir dapat dilewati oleh pengguna jalan atau tidak [4]-[8].

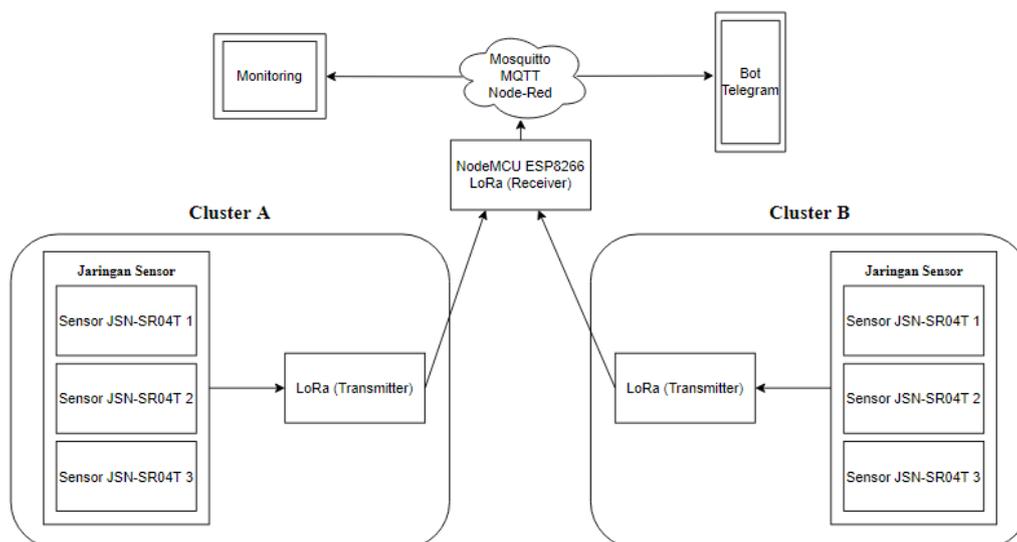
Dalam penerapan *Internet of Things* (IoT), diperlukan protokol jaringan yang berfungsi untuk menghubungkan perangkat melalui jaringan. Beberapa protokol yang mendukung penerapan IoT diantaranya *Constrained Application Protocol* (CoAP) dan MQTT [9]. Alasan utama dipilihnya protokol MQTT yang diterapkan pada perangkat IoT adalah karena jumlah penggunaan energi yang jauh lebih sedikit bila dibandingkan dengan protokol yang lain dan dapat bekerja dengan kondisi *bandwidth* kecil serta *latency* tinggi [10]. Data dengan menggunakan *platform* Thingspeak dapat digunakan pada penelitian selanjutnya [11].

Penelitian ini bertujuan untuk merancang sistem *monitoring* yang dapat membantu masyarakat dalam memantau kondisi jalan raya rawan banjir. Dengan sistem ini diharapkan dapat membantu masyarakat dalam mengambil tindakan pencegahan apabila ingin melewati jalan raya rawan banjir. Perbedaan dari sistem sebelumnya antara lain sistem ini memanfaatkan memanfaatkan modul *Long Range* (LoRa) untuk mengirimkan data sensor. Selain menampilkan data ketinggian banjir, sistem ini juga menampilkan gambar apakah jalan raya tersebut dapat dilewati oleh pejalan kaki, motor, ataupun mobil yang dapat diakses melalui *website*. Kemudian juga informasi kondisi jalan raya akan dikirimkan melalui notifikasi di aplikasi Telegram. Pada penelitian selanjutnya dapat terintegrasi dengan kendali lampu lalu lintas [12] dan menggunakan panel surya [13].

2. METODE PENELITIAN

2.1 Perancangan Sistem

Metode yang digunakan dalam penelitian ini adalah rancang bangun dalam bentuk *prototype*. Gambar 1 merupakan blok diagram sistem yang terdiri dari jaringan sensor, sistem komunikasi, dan *monitoring*. Sistem terdiri dari dua *cluster*. Proses pengiriman data dirancang menggunakan protokol MQTT. Jaringan sensor ultrasonik, LoRa, dan NodeMCU yang bertugas sebagai *publisher* akan mengirimkan data ketinggian banjir yang kemudian diteruskan ke Mosquitto yang berfungsi sebagai *broker*. Setelah data diterima oleh *broker*, kemudian data akan diteruskan kepada *subscriber* dalam hal ini adalah aplikasi IoT Node-Red. Setelah data masuk ke Node-Red, data akan diteruskan ke *dashboard monitoring* berbasis *website* dan notifikasi Telegram.



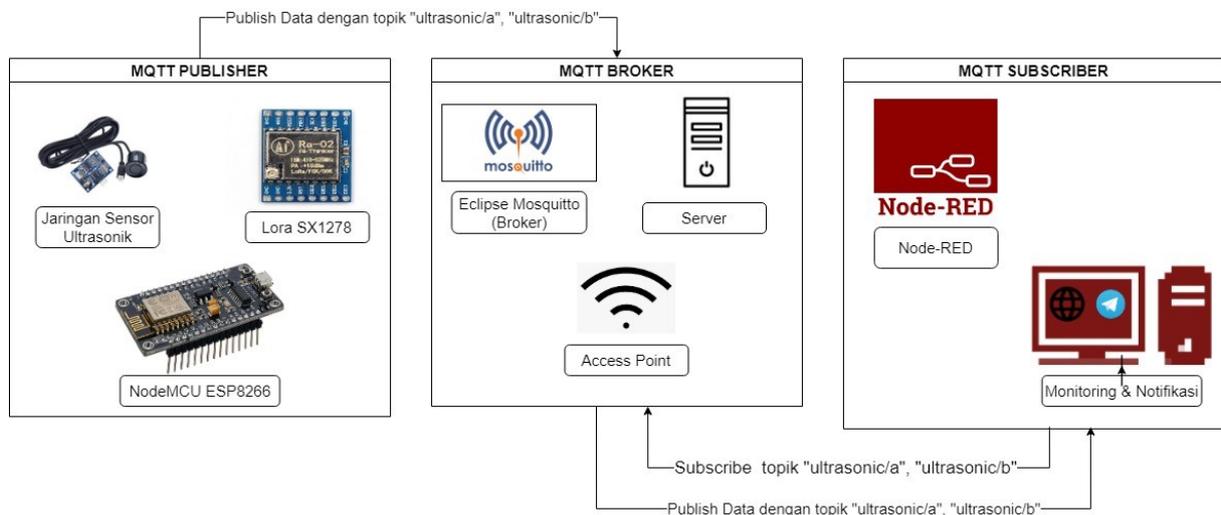
Gambar 1. Diagram blok sistem *monitoring* banjir

2.2 Arsitektur Sistem

Gambar 2 menunjukkan arsitektur IoT pada sistem yang dirancang. Jaringan sensor ultrasonik yang bertugas sebagai *publisher* akan membaca tinggi banjir dengan topik “ultrasonic/a” untuk jalan raya A dan “ultrasonic/b” untuk jalan raya B. Setelah data dibaca kemudian dikirimkan menuju LoRa yang berperan sebagai *gateway* dan diteruskan ke kontroler yaitu NodeMCU ESP8266. Selanjutnya, data akan dikirimkan melalui *broker*.

MQTT *broker* bertanggungjawab untuk mengelola aliran komunikasi antara data yang dikirimkan oleh *publisher* ke *subscriber*. Bagian ini terdiri dari komputer (*server*) yang terkoneksi dengan internet melalui *access point* dan tempat dimana *broker* beroperasi. Ketika *publisher* mengirimkan data jaringan dengan topik “ultrasonic/a” dan “ultrasonic/b”, maka *subscriber* akan menerima data jaringan sensor ultrasonik. MQTT *subscriber* bertugas untuk melakukan *subscribe* pada topik “ultrasonic/a” dan “ultrasonic/b”. Setelah mendapatkan data berupa nilai jaringan sensor ultrasonik dari *publisher*, data tersebut bisa dijadikan sistem monitor yang akan tampil pada *dashboard monitoring*.

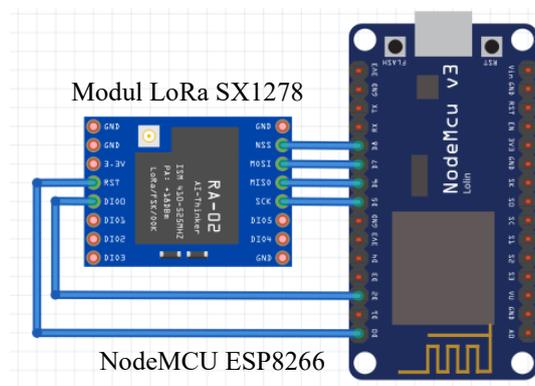
Topik pada protokol MQTT bersifat wajib. Pada setiap pengiriman, data antara *publisher* dan *subscriber* harus memiliki topik yang akan dikirimkan. Topik ini dikirimkan oleh *publisher* yaitu jaringan sensor ultrasonik, kemudian diteruskan pada *broker* berupa nilai dan diteruskan kepada *subscriber* sehingga data tersebut dapat ditampilkan pada *dashboard monitoring*.



Gambar 2. Arsitektur sistem *monitoring* banjir

2.3 Komponen Sistem

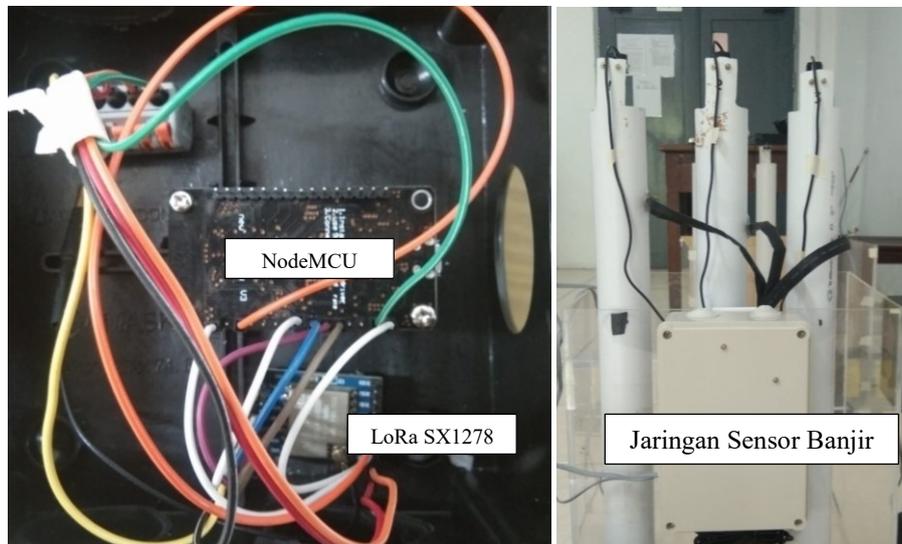
Rangkaian sistem ini memiliki beberapa *hardware* yang terdiri dari NodeMCU ESP8266, jaringan sensor ultrasonik JSN-SR04T, dan modul LoRa. Setelah rangkaian sistem dibuat, kemudian masing-masing pin dikonfigurasi seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Skematik sistem modul LoRa dan NodeMCU ESP8266

3. HASIL DAN PEMBAHASAN

Implementasi sistem dilakukan sesuai dengan diagram skematik yang dijelaskan di bagian metode penelitian. Adapun hasil implementasi dari sistem *hardware* dapat dilihat pada Gambar 4. Setelah dilakukan integrasi jaringan sensor, LoRa, dan NodeMCU ESP8266 yang akan digunakan untuk mem-*publish* data. Langkah selanjutnya adalah pengujian LoRa dan pengiriman data menggunakan protokol MQTT.



Gambar 4. Realisasi sistem *monitoring* banjir

3.1 Pengujian Modul LoRa SX1278

Modul LoRa SX1278 memberikan data berupa nilai *Received Signal Strength Indicator* (RSSI) dengan frekuensi 433 MHz yang mengirimkan sebuah informasi secara nirkabel. Pengujian pada modul LoRa SX1278 ini untuk mengetahui seberapa jauh jangkauan dalam mengirimkan data dari LoRa *transmitter* ke *receiver*. Hasil pengujian jangkauan modul LoRa SX1278 dapat dilihat pada Tabel 1.

Tabel 1. Pengujian jangkauan modul LoRa SX1278

No.	Jarak (m)	RSSI (dBm)
1	10	-116
2	20	-120
3	30	-120
4	40	-126
5	50	-128
6	60	-128
7	70	-130
8	80	-133
9	90	-133
10	100	-139
11	150	-141
12	200	-146
13	250	-150
14	300	-153
15	350	-159
16	> 350	-

Dari pengujian didapatkan kesimpulan bahwa semakin jauh jaraknya, semakin kecil nilai RSSI. Jarak maksimal yang diuji coba adalah 400 meter dan mengalami kegagalan pada jarak di atas 350 meter.

3.2 Pengujian Koneksi dan Komunikasi

Pengujian ini bertujuan untuk mengetahui batas jarak maksimal koneksi *hotspot* dapat terhubung serta apakah komunikasi MQTT dapat berjalan dengan baik. Pengujian *hotspot* dilakukan dengan menghubungkan purwarupa *node* dengan *access point*. Pengujian komunikasi MQTT dilakukan dengan menghitung waktu pesan (*payload*) yang dikirimkan *publisher* untuk sampai kepada *subscriber*. Hasil pengujian jarak maksimum *access point* dapat dilihat pada Tabel 2.

Tabel 2. Pengujian jarak maksimum *access point*

No.	Jarak <i>access point</i> (m)	Status koneksi
1	3	Terhubung
2	6	Terhubung
3	9	Terhubung
4	12	Terhubung
5	15	Terhubung
6	18	Terhubung
7	21	Terhubung
8	24	Terhubung
9	27	Terhubung
10	30	Terhubung
11	33	Tidak terhubung
12	36	Tidak terhubung

Dari pengujian didapatkan kesimpulan bahwa sistem dapat terkoneksi dengan *hotspot* sampai dengan jarak 30 meter. Untuk jarak lebih dari 30 meter sistem sudah tidak dapat terkoneksi dengan jaringan internet berbasis *hotspot*. Hasil pengujian komunikasi MQTT dapat dilihat pada Tabel 3.

Tabel 3. Pengujian komunikasi MQTT

Pengiriman ke-	Waktu <i>publish</i> (NodeMCU ESP8266)	Waktu <i>subscribe</i> (Node-Red MQTT-in)	<i>Delay</i> waktu pengiriman (s)
1	12:24:43.630	12:24:43.680	0,050
2	12:24:57.932	12:24:57.940	0,008
3	12:25:09.481	12:25:09.512	0,031
4	12:25:20.735	12:25:20.774	0,039
5	12:25:29.774	12:25:29.783	0,009
6	12:25:44.054	12:25:44.109	0,055
7	12:26:01.880	12:26:01.935	0,055
8	12:26:11.819	12:26:11.863	0,044
9	12:26:23.113	12:26:23.183	0,070
10	12:26:32.316	12:26:32.344	0,028
11	12:26:45.540	12:26:45.562	0,022
12	12:26:53.201	12:26:53.250	0,049
13	12:27:07.908	12:27:07.972	0,064
14	12:27:19.232	12:27:19.268	0,036
15	12:27:26.907	12:27:26.942	0,035
16	12:27:37.904	12:27:37.949	0,045
17	12:27:50.661	12:27:50.696	0,035
18	12:28:08.755	12:28:08.826	0,071
19	12:28:20.081	12:28:20.136	0,055
20	12:28:31.453	12:28:31.492	0,039
Rata-rata			0,042

Dari pengujian didapatkan kesimpulan nilai *payload* yang dikirimkan dari NodeMCU ESP8266 dengan jaringan sensor JSN-SR04T (*publish*) memiliki rata-rata selang waktu pengiriman selama 0,042 detik.

3.3 Pengujian *Dashboard Monitoring* pada Node-Red

Pada pengujian ini digunakan *node* MQTT-in sebagai nilai *input* pesan (*payload*), *node Scalable Vector Graphics* (SVG), *node text*, dan *node chart* sebagai *output*, dan dua jenis *node function* yang berguna untuk memodifikasi nilai *payload*. Berikut tampilan *flow node* pengujian pada Node-Red dapat dilihat pada Gambar 5.



Gambar 5. *Flow node* pengujian *dashboard* pada Node-Red

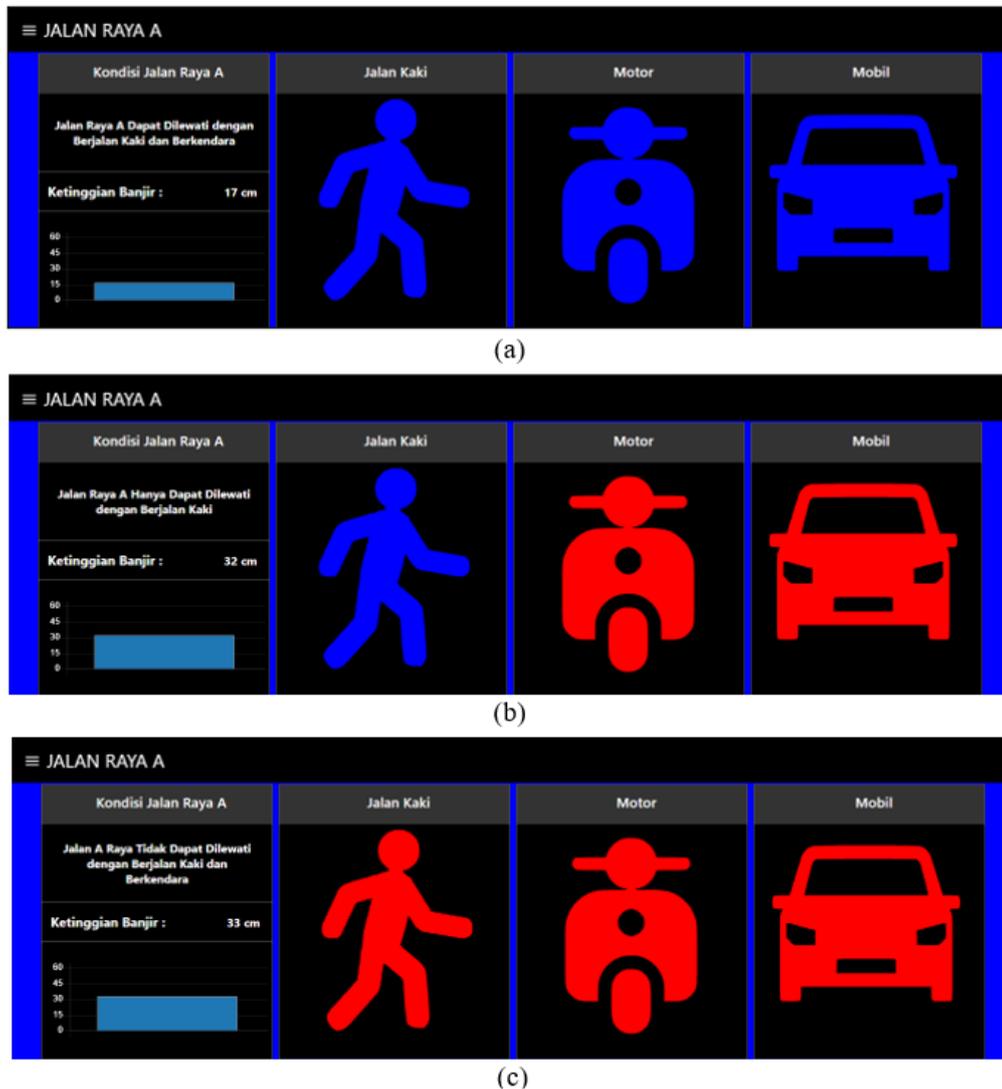
Pada pengujian ini, pesan yang bernilai angka akan diubah menjadi pesan bertipe *string* yang berisi kalimat apakah jalan raya tersebut dapat dilewati atau tidak oleh pejalan kaki, motor, dan mobil dan pesan bertipe *JavaScript Object Notation* (JSON) yang berfungsi untuk mengubah atribut warna gambar SVG. Hasil pengujian *dashboard monitoring* pada Node-Red dapat dilihat pada Tabel 4, Tabel 5, dan Gambar 6.

Tabel 4. Hasil pengujian *dashboard monitoring* pada Node-Red

No.	Input MQTT	Output		
		Ketinggian banjir (cm)	Bar chart (cm)	Kondisi jalan raya
1	14	14	14	Jalan Raya Dapat Dilewati dengan Berjalan Kaki dan Berkendara
2	17	17	17	Jalan Raya Dapat Dilewati dengan Berjalan Kaki dan Berkendara
3	28	28	28	Jalan Raya Hanya Dapat Dilewati dengan Berjalan Kaki
4	32	32	32	Jalan Raya Hanya Dapat Dilewati dengan Berjalan Kaki
5	33	33	33	Jalan Raya Tidak Dapat Dilewati dengan Berjalan Kaki dan Berkendara
6	50	50	50	Jalan Raya Tidak Dapat Dilewati dengan Berjalan Kaki dan Berkendara

Tabel 5. Hasil pengujian *dashboard monitoring* pada Node-Red lanjutan

No.	Input MQTT	Output		
		Pengguna jalan		
		Gambar SVG pejalan kaki	Gambar SVG motor	Gambar SVG mobil
1	14	Biru	Biru	Biru
2	17	Biru	Biru	Biru
3	28	Biru	Merah	Merah
4	32	Biru	Merah	Merah
5	33	Merah	Merah	Merah
6	50	Merah	Merah	Merah



Gambar 6. Percobaan pengujian *dashboard monitoring* pada Node-Red: (a) status jalan raya dapat dilewati dengan berjalan kaki dan berkendara, (b) status jalan raya hanya dapat dilewati dengan berjalan kaki, (c) status jalan raya tidak dapat dilewati dengan berjalan kaki dan berkendara

Berdasarkan dari hasil pengujian *input/output dashboard* Node-Red pada Tabel 4, Tabel 5, dan Gambar 6 didapatkan bahwa nilai pesan *output* ketinggian banjir dan diagram batang akan bernilai sama dengan nilai *input*. Pada kondisi (a), pesan *input* yang bernilai kurang dari sama dengan 17 akan menghasilkan *output* kondisi jalan raya berupa kalimat bahwa jalan raya dapat dilewati oleh semua pengguna jalan dan semua gambar SVG berwarna biru. Pada kondisi (b), pesan yang bernilai antara 18 sampai dengan 32 akan menghasilkan *output* kondisi jalan raya berupa kalimat bahwa jalan raya hanya dapat dilewati oleh pejalan kaki dan gambar SVG motor dan mobil berubah menjadi berwarna merah. Sedangkan pada kondisi (c), pesan yang bernilai lebih dari sama dengan 33 akan menghasilkan *output* kondisi jalan raya berupa kalimat bahwa jalan raya tidak dapat dilewati oleh semua pengguna jalan dan semua gambar SVG berwarna merah.

3.4 Pengujian *Bot Telegram*

Pengujian dilakukan dengan menggunakan aplikasi Node-Red sebagai media transmisi pengiriman data ketinggian banjir dan *bot Telegram* sebagai media notifikasi. Digunakan *node MQTT-in* sebagai nilai *input* pesan (*payload*), *node Telegram sender* sebagai *output*, dan dua jenis *node function* yang berguna untuk memodifikasi nilai *payload*. Berikut tampilan *flow node* pengujian *bot Telegram* pada Node-Red dapat dilihat pada Gambar 7.

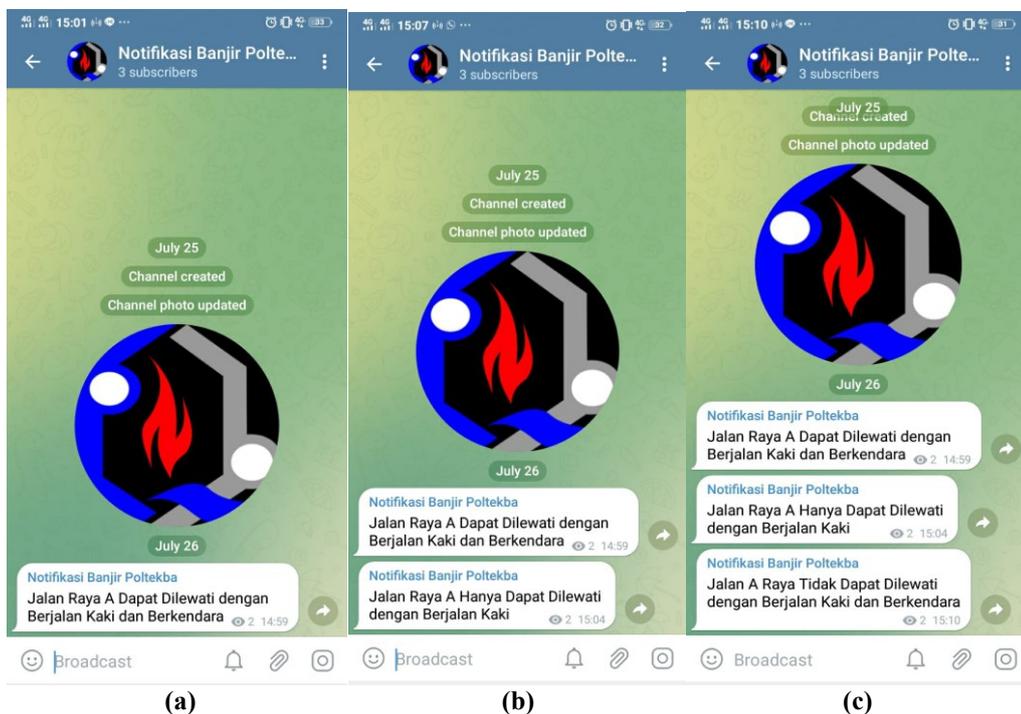


Gambar 7. Percobaan pengujian *dashboard monitoring* pada Node-Red

Pada pengujian ini, *node function* berfungsi untuk menentukan alamat penerima dan mengirimkan pesan notifikasi. Sedangkan *node Telegram sender* berfungsi menghubungkan Node-Red dengan *bot* Telegram. Hasil pengujian *bot* Telegram dapat dilihat pada Tabel 6 dan Gambar 8.

Tabel 6. Hasil pengujian *bot* Telegram

No.	Input MQTT	Output		
		Ketinggian banjir (cm)	Bar chart (cm)	Kondisi jalan raya
1	14	14	14	Jalan Raya Dapat Dilewati dengan Berjalan Kaki dan Berkendara
2	17	17	17	Jalan Raya Dapat Dilewati dengan Berjalan Kaki dan Berkendara
3	28	28	28	Jalan Raya Hanya Dapat Dilewati dengan Berjalan Kaki
4	32	32	32	Jalan Raya Hanya Dapat Dilewati dengan Berjalan Kaki
5	33	33	33	Jalan Raya Tidak Dapat Dilewati dengan Berjalan Kaki dan Berkendara
6	50	50	50	Jalan Raya Tidak Dapat Dilewati dengan Berjalan Kaki dan Berkendara



Gambar 8. Percobaan pengujian *bot* Telegram untuk tiga status yang berbeda: (a) status jalan raya dapat dilewati dengan berjalan kaki dan berkendara, (b) status jalan raya hanya dapat dilewati dengan berjalan kaki, (c) status jalan raya tidak dapat dilewati dengan berjalan kaki dan berkendara

Percobaan pengujian berdasarkan dari hasil pengujian *bot* Telegram pada Node-Red pada Tabel 6 dan Gambar 8 didapatkan bahwa pada kondisi (a) nilai pesan *input* yang bernilai kurang dari sama dengan 17 akan menghasilkan *output* berupa notifikasi Telegram berisi “Jalan Raya Dapat Dilewati

dengan Berjalan Kaki dan Berkendara.” Pada kondisi (b), pesan yang bernilai antara 18 sampai dengan 32 akan menghasilkan *output* berupa notifikasi Telegram “Jalan Raya Hanya Dapat Dilewati dengan Berjalan Kaki.” Sedangkan pada kondisi (c), pesan yang bernilai lebih dari sama dengan 33 akan menghasilkan *output* berupa notifikasi Telegram “Jalan Raya Tidak Dapat Dilewati dengan Berjalan Kaki dan Berkendara.”

4. KESIMPULAN

Sistem *monitoring* pada jaringan sensor banjir jalan raya menggunakan protokol MQTT telah berhasil dirancang. Berdasarkan hasil pengujian, NodeMCU ESP8266 dapat terhubung ke jaringan internet dengan jarak maksimal 30 meter dari *access point*. LoRa sebagai *gateway* dapat mengirimkan dan menerima paket data dari jaringan sensor sampai dengan jarak 100 meter. Pada pengujian *delay* dalam menghitung lama proses dari mulai awal data sensor di-*publish* hingga data sampai di *subscriber* didapatkan bahwa waktu yang dibutuhkan untuk satu kali pengiriman data membutuhkan waktu rata-rata 0,042 detik. Berdasarkan pengujian sistem dengan memanfaatkan aplikasi Node-Red sebagai penghubung *node* didapatkan hasil berupa berjalannya seluruh fungsi yang ada dalam sistem sesuai dengan fungsinya. Dengan sistem ini, data dari jaringan sensor dapat ditampilkan pada *dashboard monitoring* atau *subscriber* secara *real-time* dan memberikan notifikasi melalui *bot* Telegram. Pada penelitian berikutnya diharapkan dapat meningkatkan kualitas alat yang digunakan sehingga jarak yang dijangkau oleh alat dapat bekerja lebih jauh.

REFERENSI

- [1] J. U. D. Hatmoko, B. H. Setiadji, and M. A. Wibowo, “Evaluasi Pengaruh Banjir, Beban Berlebih, Dan Mutu Konstruksi Pada Kondisi Jalan,” *Jurnal Transportasi*, vol. 17, pp. 89-98, 2017.
- [2] J. P. Nainggolan, M. E. I. Najoan, and S. D. S. Karouw, “Pengembangan Sistem Informasi Peringatan Dini Banjir di Kota Manado Berbasis Internet of Things,” *Jurnal Teknik Informatika*, vol. 15, no. 1, pp. 65-74, 2020.
- [3] A. R. Alfaridzi, E. Kurniawan, and A. Sugiana, “IoT Blynk Untuk Sistem Monitoring Pendeteksi Dini Banjir Sungai Citarum Terintegrasi Media Sosial,” *eProceedings of Engineering*, vol. 7, no. 1, pp. 43-52, 2020.
- [4] D. R. Ariyani and R. E. Putri, “Sistem Monitoring Banjir Pada Jalan Menggunakan Aplikasi Mobile Dan Modul Wi-Fi,” in *Prosiding SEMNASTEK 2017*, Jakarta, 2017, pp. 1-8.
- [5] M. A. Y. Bustomi and Asmunin, “Rancang Bangun Sistem Monitoring Peringatan Dini Bencana Banjir Berbasis Iot Menggunakan Protokol MQTT Dengan Notifikasi Bot Telegram,” *Jurnal Manajemen Informatika*, vol. 12, no. 1, pp. 1-13, 2021.
- [6] A. Firmansyah, A. Sasongko, and M. A. Said, “Sistem Monitoring Ketinggian Air Dan Notifikasi Peringatan Banjir Pada Pintu Air Berbasis Android,” *Tugas Akhir Politeknik Harapan Bersama Tegal*, 2020.
- [7] R. Ningsih, I. A. Bangsa, and A. Stefanie, “Perancangan Sistem Monitoring dan Pendeteksi Banjir Menggunakan Metode Background Subtraction Berbasis Internet Of Things (IoT),” *Jurnal Teknik Elektro dan Vokasional*, vol. 5, no. 1, pp. 97-100, 2019.
- [8] A. Tenggono, Y. Wijaya, E. Kusuma, and Welly, “Sistem Monitoring dan Peringatan Ketinggian Air Berbasis Web dan SMS Gateway,” *Jurnal Ilmiah SISFOTENIKA*, vol. 5, no. 2, pp. 119-129, 2015.
- [9] D. Bilal, A. U. Rehman, and R. Ali, “Internet of Things (IoT) Protocols: A Brief Exploration of MQTT and CoAP,” *International Journal of Computer Applications*, vol. 179, no. 27, pp. 9-14, 2018.
- [10] S. M. Kim, H. S. Choi, and W. S. Rhee, “IoT Home Gateway for Auto-Configuration and Management of MQTT Devices,” in *IEEE Conference on Wireless Sensor (ICWISE)*, Malacca, 2015, pp. 12–17.
- [11] E. Sorongan and Q. Hidayati, “ThingSpeak Sebagai Sistem Monitoring Tangki Spbu Berbasis Internet Of Things E Sorongan”, *JTERA (Jurnal Teknologi Rekayasa)*, vol. 3, no. 2, pp. 219-224, 2018.
- [12] Q. Hidayati, “Kendali Lampu Lalu Lintas Dengan Deteksi Kendaraan Menggunakan Metode Blob Detection”, *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 6, no. 2, pp. 215-221, 2017.
- [13] Q. Hidayati and N. Yanti, “Sistem Pembangkit Panel Surya Dengan Solar Tracker Dual Axis,” in *Prosiding SNITT Poltekba*, Balikpapan, 2020, pp. 68-73.

