

PROTOTYPE SISTEM PENDETEKSI BANJIR MENGGUNAKAN NODEMCU DAN PROTOKOL MQTT BERBASIS INTERNET OF THINGS

Sutarti¹, Anharudin², Syukron Rosadi³

^{1,2,3}Program Studi Sistem Komputer, Fakultas Teknologi Informasi Universitas Serang Raya Jl.
Raya Serang – Cilegon Taman Drangong Serang, Indonesia

e-mail: *¹sutarti86@gmail.com, ²anhar.dean@gmail.com, ³rosadisyyukron@gmail.com

Abstract

Flood disasters that often occur around the Pamarayan Dam were caused by heavy rainfall and the flow of river water in the Lebak and Bogor areas, causing the overflow of the water discharge in the dam. In addition to causing property losses, floods can also cause casualties. Therefore we need a flood detection and warning system to avoid casualties and minimize material and psychological losses. The system was built using the Internet of Things and the Message Queue Telemetry Transport (MQTT) protocol. This flood detection system is also designed to be able to deliver warnings quickly and with wide coverage. This flood detection system uses ultrasonic sensors, NodeMCU, LED, and buzzer. The ultrasonic sensor is used to measure the water level which data can be read by the NodeMCU as a data transmission medium from the sensor to the MQTT server. With this system, the water level can be monitored via the cayenne and send notifications to telegram. The LED will be green if the water is in Safe status, yellow if the alert status, and red if the danger status is accompanied by a warning alarm.

Keyword: Buzzer, Cayenne, Internet of Things, MQTT, Notification.

PENDAHULUAN

Bencana banjir sering terjadi dan masih menjadi salah satu fokus perhatian. Pasalnya bencana banjir itu mengakibatkan banyak korban jiwa, serta juga menimbulkan banyak kerugian, baik kerugian material maupun psikologis. Bencana banjir yang sering terjadi nampak tidak ada pencegahan secara efektif untuk meminimalisir korban jiwa, serta juga masih minimnya sistem untuk memberi peringatan sedini mungkin akan datangnya banjir agar kerugian bisa dikurangi. Penggunaan berbagai macam sensor dan teknologi sudah lama banyak dikembangkan untuk memonitor kondisi lingkungan dan bencana, contohnya penggunaan alat deteksi banjir menggunakan *Radar doppler*, tetapi masih memerlukan rancangan perangkat keras yang rumit dan memerlukan biaya yang cukup besar (Umari, Anggraini, dan Muttaqin, 2017).

Internet of Things (IoT) adalah teknologi kendali atau *monitoring* jarak jauh yang memanfaatkan jaringan *internet* sebagai penghubungnya. Pada umumnya IoT menggunakan *gadget* atau *android* sebagai media monitoringnya sehingga juga mempermudah *user* untuk pengoperasiannya (Abilovani, Widhi, dan Bakhtiar, 2018). Kemudian pada sistem *monitoring* untuk perangkat IoT dilakukan pengembangan yang memakai protokol MQTT (*Message Queue Telemetry Transport*). Protokol ini dinilai sesuai dengan perangkat IoT yang bersifat *lightweighted message* serta dirancang pada perangkat dengan sumber daya *minimum*. Prinsip *publish/subscribe* yang dimiliki protokol MQTT dan dapat beradaptasi dalam mengirim dan menerima pesan dalam *monitoring* yang berdasarkan dengan topik yang dikehendaki. Alasan utama dipilihnya protokol MQTT yang diterapkan pada perangkat IoT ialah karena jumlah penggunaan energi yang jauh lebih sedikit bila dibandingkan dengan protokol yang lain dan dapat bekerja di kehidupan (Saputra, Afrizal, Mahfud, dan Pamungkas, 2017).

Bendungan Pamarayan yang berlokasi di Kabupaten Serang, Provinsi Banten selesai dibangun pada tahun 1997 merupakan pengganti dari bendung lama yang dibangun pada zaman

pemerintahan Belanda. Bencana banjir yang sering terjadi di sekitar bendungan pamarayan diakibatkan karena curah hujan yang lebat dan juga adanya kiriman air sungai yang ada di daerah lebak dan bogor sehingga menyebabkan meluapnya debit air yang ada pada bendungan. Ditahun 2015-2018 terjadi banjir kurang lebih setinggi 2 meter di sekitaran bendungan pamarayan yang mengakibatkan 170 rumah terendam air yang berada di 3 desa diantaranya desa pamarayan, cikeusal, dan tengkurak yang menyebabkan banyaknya kerugian secara material maupun psikologis, banyak masyarakat yang terkena dampak banjir mengeluh karena tidak adanya informasi ketinggian air sebelum terjadinya banjir. Maka dibutuhkan alat untuk mendeteksi ketinggian air seperti sensor, yang mana berguna untuk mendeteksi jarak ketinggian air sehingga dapat mengetahui seberapa tinggi air tersebut meluap.

Sistem peringatan dini banjir dapat dibangun menggunakan sensor ultrasonik dan mikrokontroler (Umari et al., 2017), atau bisa memanfaatkan arduino uno (Akhiruddin, 2018). Teknologi IoT dan WSN dapat memberikan informasi monitoring banjir secara realtime (Patil, et al., 2019). Informasi lokasi banjir dapat dikirim melalui jaringan GSM (Efendi dan Wildian, 2018).

Protokol MQTT dapat diimplementasikan untuk sistem monitoring perangkat IoT (Abilovani et al., 2018). Protokol MQTT dapat digunakan pada sistem monitoring banjir (Hasiholan et al., 2018). Protokol MQTT dapat diimplementasikan dengan menggunakan topologi WAN (Saputra, et al., 2017). Alat pemantau jumlah dan biaya pemakaian air berbasis android menggunakan sensor water flow meter, LCD, Arduino Nano dan NodeMCU yang dapat menampilkan data melalui LCD dan android (Hakim, et al., 2018).

IoT dapat digunakan untuk kontrol lampu menggunakan Arduino berbasis web (Artono dan Putra, 2017). Sistem dapat memonitor dan dapat difungsikan untuk pengendali lampu dengan project web yang dibuat di aplikasi dari web dan aplikasi cayenne di smartphone dapat diketahui apakah kondisi lampu dalam keadaan hidup atau mati. Siswanto et.al. (2020) membuat *prototype smart home* berbasis IoT menggunakan NodeMCU dan Telegram. Pada sistem *smart home* berkonsep IoT ini aman karena hanya orang-orang yang memiliki akses yang dapat mengontrol rumah seperti membuka kunci pintu dan menyalakan lampu dari jarak jauh.

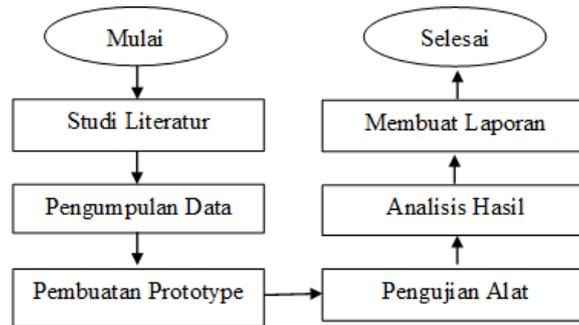
Rizki (2018) mengimplementasikan dan menguji Modul Esp8266 dengan Aplikasi Android MQTT-Dash pada Jaringan MQTT. Dari Penelitian tersebut dapat disimpulkan bahwa sistem dengan protokol MQTT dapat berjalan dengan baik. Dengan metode yang telah diterapkan, Esp8266 tidak mengalami masalah walaupun berjalan selama 10 hari.

Sistem pada penelitian ini menggunakan komponen sensor ultrasonik dan NodeMCU untuk mengukur jarak ketinggian permukaan air dan data hasil pembacaan akan ditampilkan pada layar *computer* melalui aplikasi *Cayenne* dan sistem akan mengirimkan sebuah notifikasi berupa pesan *Telegram* yang berada di *smartphone*. Sistem ini bermanfaat untuk memudahkan masyarakat mengetahui informasi banjir sebelum banjir datang agar masyarakat bisa siap siaga dan juga bisa mengamankan barang berharga agar tidak banyak kerugian yang dialami.

METODE PENELITIAN

Tahapan Penelitian

Penelitian ini diawali dengan melakukan studi pendahuluan, yaitu tahap awal dalam metodologi penelitian ini. Di tahap ini dilakukan studi lapangan dengan terjun langsung ke bendungan. Pengamatan langsung dilakukan dengan tujuan mengetahui informasi-informasi awal mengenai bendungan dalam hal ini, secara jelas dapat memahami mengenai seberapa tinggi volume air dan rawan bencana banjir pada bendungan Pamarayan. Berikut tahapan penelitian yang akan dilakukan dapat dilihat pada gambar 1.



Gambar 1. Diagram alur penelitian

Alat dan Bahan Penelitian

Adapun alat-alat yang diperlukan untuk penelitian ini yaitu:

1. *Laptop* (Untuk memprogram NodeMCU)
2. *HP android* (yang sudah ter-*install* apk Cayenne)
3. *Bread board*
4. NodeMCU
5. Kabel USB
6. Sensor *Ultrasonik*
7. *Adaptor 5V*
8. LED
9. *Buzzer*
10. *Software*
 - a). *Arduino IDE 1.0.5*
 - b). *Aplikasi Cayenne*

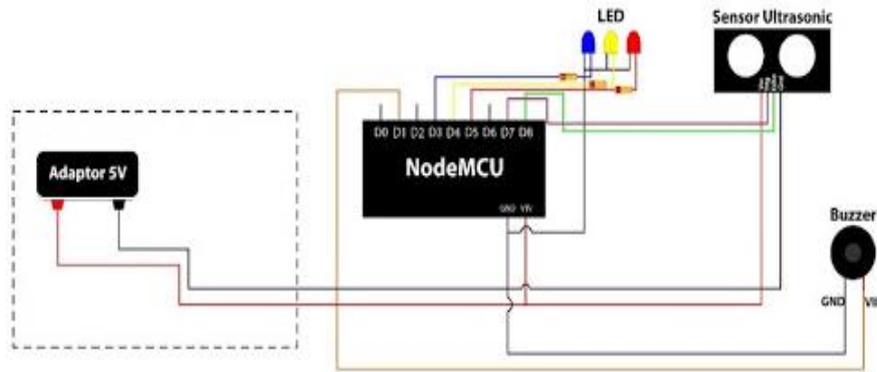
Adapun bahan-bahan yang diperlukan untuk penelitian ini yaitu data ketinggian air yang ada di Bendungan Pamarayan ini dari tahun 2014-2019.

Tabel 1. Data Ketinggian dan Debit air

No	Tgl/Tahun	Ketinggian Air (Meter)	Debit Air (m ³ /detik)	Status
1	12-12-2014	2,20	798	Siaga Banjir
2	02-07- 2015	3,67	1.418	Siaga 1
3	08-01-2015	3,94	2.300	Siaga 1
4	10-01-2016	2,43	2.600	Siaga Banjir
5	21-01-2017	3,75	798	Siaga 1
6	05-07-2017	4,20	1.200	Banjir
7	26-04-2018	5,10	926	Banjir
8	17-12-2018	5,40	1.300	Banjir
9	22-01-2019	4,30	908	Banjir

Deskripsi Usulan Penelitian

Penelitian ini bertujuan untuk mendeteksi ketinggian air secara *online* sebagai informasi dini terhadap terjadinya banjir. Sistem pendeteksi banjir ini menggunakan pendekatan teknologi *Internet of things* (IoT) agar informasi ketinggian air dapat diketahui secara *real time*.

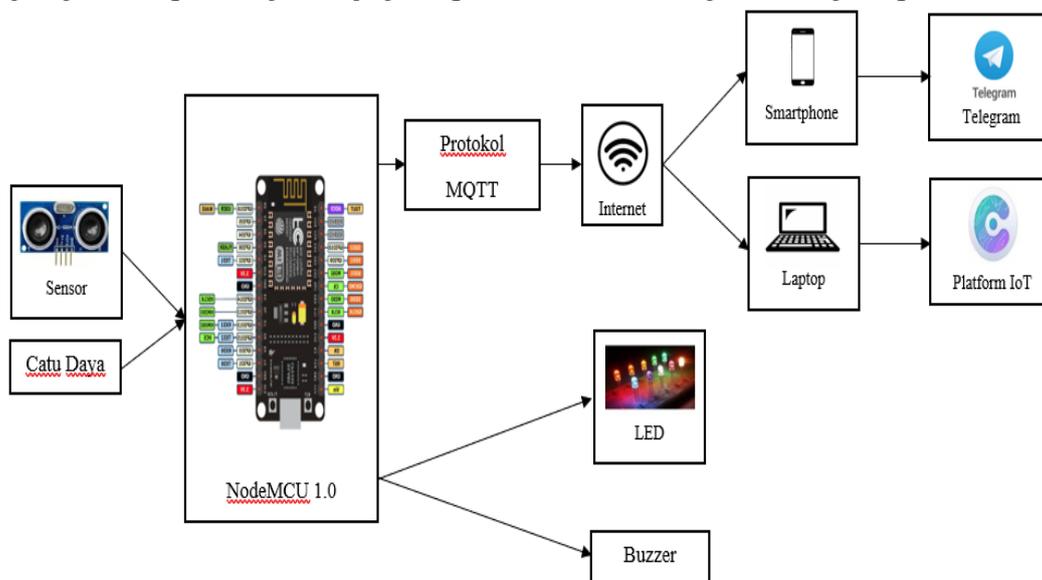


Gambar 2. Skema Rangkaian *Prototype*

Sensor ultrasonik digunakan sebagai pembaca data dan NodeMCU sebagai pemroses dan mengirimkan data secara *nirkabel* ke aplikasi *cayenne* yang sudah terinstal di *smartphone android* lewat *broker* yang ada di *server MQTT cayenne*.

Perancangan perangkat keras

Perangkat keras yang dibutuhkan pada pembuatan *prototype* ini terdiri dari sensor ultrasonik, *mikrokontroler* NodeMCU/Esp8266, *LED*, *buzzer*, serta bahan pendukung lainnya. Adapun perancangan tahapan kerja perangkat keras tersebut dapat dilihat pada gambar 3.



Gambar 3. Blok Diagram Perancangan Alat

Perancangan perangkat lunak (*Software*)

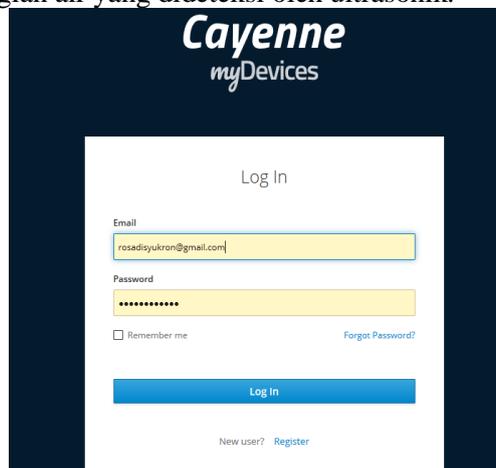
Tahap berikutnya merupakan tahap pembuatan perangkat lunak (*software*). Pembuatan perangkat lunak dibagi menjadi dua tahap, yaitu mempersiapkan *webserver* IoT dan pembuatan program *prototype*. *Webserver* IoT merupakan syarat utama pembuatan perangkat IoT, karena perangkat IoT membutuhkan *internet* sebagai media penghubung antar perangkat. *Webserver* IoT lebih dikenal sebagai *platform* IoT telah banyak beredar di kalangan masyarakat.

Sebelum melakukan perancangan dan pemrograman alat langkah pertama yang harus dilakukan adalah menginstal suatu *software Arduino IDE* dan *library* untuk melakukan pemrograman alat:

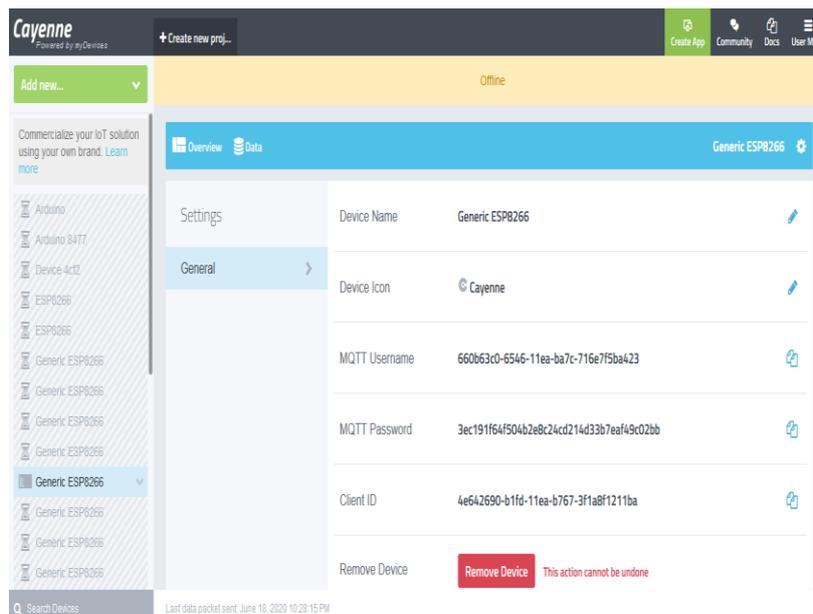
1. Setelah *file* instalasi dijalankan maka akan muncul jendela *License Agreement* lalu klik tombol *I Agree*.
2. Berikutnya akan diminta memasukkan *folder* instalasi *Arduino*. Biarkan *default* C:\Program Files\Arduino.

3. Lalu akan ada tampilan jendela ‘*Setup Installation Options*’ sebaiknya dicentang semua opsinya.
4. Proses Instalasi akan dimulai.
5. Ditengah proses instalasi, jika komputer belum terinstal *driver USB* maka akan muncul ‘*Security Warning*’ pilih saja tombol instal.
6. Tunggu sampai proses instalasi ‘*Completed*’
7. Pada tahap ini *Software Arduino IDE* sudah terinstal.

Program *prototype* dibuat berdasarkan gabungan *code source* MQTT *cayenne*, sensor *ultrasonik*, *Mikrokontroler* NodeMCU lampu *led* dan *buzzer* yang ada pada *library* *Arduino*. Langkah pertama sebelum mengakses *platform* IoT *cayenne* adalah harus mendaftarkan diri terlebih dahulu dengan cara mengisi nama, alamat *Email* dan *Password*. Setelah mendaftarkan diri akan diarahkan untuk *login* menggunakan alamat *Email* dan *Password* yang sudah didaftarkan agar bisa masuk ke *dashboard*. Setelah pendaftaran selesai, maka proses berikutnya adalah pemilihan perangkat *mikrokontroler* NodeMCU. Kemudian *cayenne* akan memberikan MQTT *username*, MQTT *password*, dan *client ID* *platform*, sebagai alamat *platform* yang akan di masukan ke dalam program untuk mengkoneksikan NodeMCU ke *cayenne* agar bisa menampilkan jarak ketinggian air yang dideteksi oleh ultrasonik.

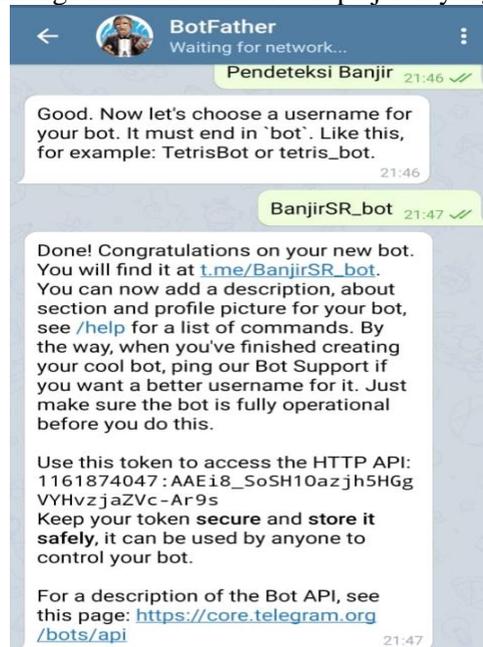


Gambar 4. Tampilan Log-in My Device Cayenne



Gambar 5. MQTT Username, MQTT Password dan Client ID

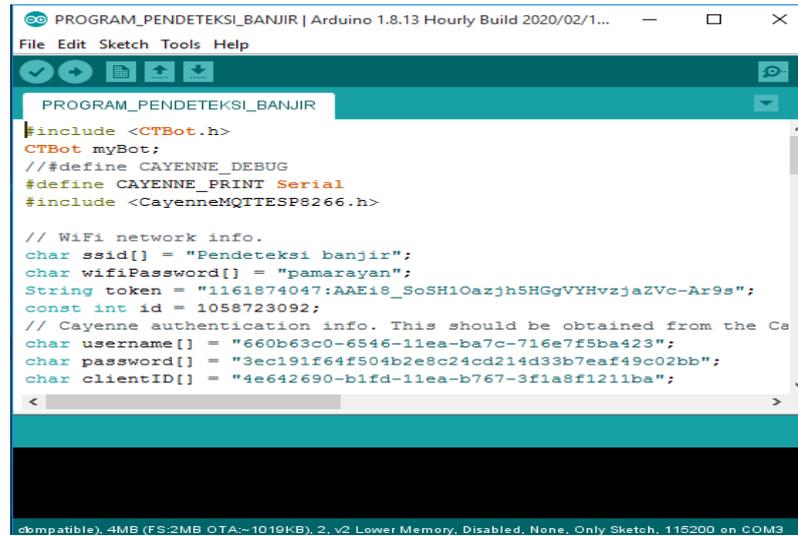
Langkah selanjutnya adalah konfigurasi Telegram. Pada tahap ini yang dilakukan adalah men-download aplikasi *telegram*, kemudian mencari *BotFather* untuk mendaftarkan *bot telegram* dengan cara mengisi nama sesuai yang diinginkan. Setelah selesai mendaftarkan nama *bot telegram* pada *BotFather* maka akan diberikan *token* untuk akses ke *HTTP API*, *token* ini nantinya berguna untuk dituliskan pada program *Arduino ide* untuk mengkoneksikan *NodeMCU* dengan *telegram* agar bisa mengirimkan notifikasi berupa jarak yang dideteksi oleh sensor.



Gambar 6. Tampilan Pesan Token Bot Telegram

Mikrokontroler NodeMCU diprogram menggunakan *Arduino IDE*. Sebelumnya menulis program, *install library* yang dibutuhkan di *software Arduino IDE* agar bisa mendukung *board* NodeMCU, pengaturan yang dibuat adalah sebagai berikut:

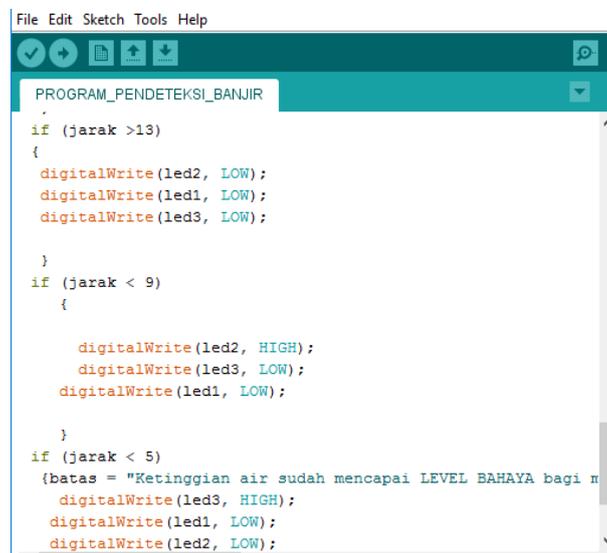
1. Buka bagian *File Prefences*, di kotak *additional Board Manage URL Input Link*: [//arduino.esp8266.com/stable/package_esp8266com_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json), atau bisa dicari di www.blynk.cc.
2. Selanjutnya buka bagian *Tools* ke *Manage libraries*, cari "NodeMCU 1.0", pada daftar akan muncul "NodeMCU 1.0 (ESP-12E Module)" *install* versi terakhirnya. *Board* ini untuk memprogram NodeMCU.
3. Selanjutnya buka bagian *Tools* ke *Manage libraries*, cari "CTBot", pada daftar akan muncul "CTBot" *install* versi terakhirnya.
4. Selanjutnya buka bagian *Tools* ke *Manage libraries*, cari "Cayenne", pada daftar akan muncul "Cayenne" *install* versi terakhirnya.
5. Selanjutnya pengaturan pada bagian *Tools Board* sorot dan pilih "NODEMCU V1.0 (ESP8266)".
6. Pada bagian *port* pilih "COM3/COM5".
7. Tulis kode pemrograman pada halaman yang disediakan (kode pemrograman pada *system* ini disertakan di halaman Lampiran).
Upload/flash kode yang telah dibuat, jika terdapat keterangan *done uploading* maka proses pemrograman mikrokontroler dengan *Arduino IDE* berhasil.



```
PROGRAM_PENDETEKSI_BANJIR
#include <CTBot.h>
CTBot myBot;
// #define CAYENNE_DEBUG
#define CAYENNE_PRINT Serial
#include <CayenneMQTTESP8266.h>

// WiFi network info.
char ssid[] = "Pendeteksi banjir";
char wifiPassword[] = "pamarayan";
String token = "1161874047:AAE18_SoSH1Oazjh5HGgVYHvzja2Vc-Ar9s";
const int id = 1058723092;
// Cayenne authentication info. This should be obtained from the Cayenne
char username[] = "660b63c0-6546-11ea-ba7c-716e7f5ba423";
char password[] = "3ec191f64f504b2e8c24cd214d33b7eaf49c02bb";
char clientID[] = "4e642690-b1fd-11ea-b767-3f1a8f1211ba";
```

Gambar 7. Kode Program Bot Telegram, Wifi dan Cayenne



```
PROGRAM_PENDETEKSI_BANJIR
if (jarak > 13)
{
digitalWrite(led2, LOW);
digitalWrite(led1, LOW);
digitalWrite(led3, LOW);
}
if (jarak < 9)
{
digitalWrite(led2, HIGH);
digitalWrite(led3, LOW);
digitalWrite(led1, LOW);
}
if (jarak < 5)
{
(batas = "Ketinggian air sudah mencapai LEVEL BAHAYA bagi m
digitalWrite(led3, HIGH);
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
```

Gambar 8. Kode Program untuk Menentukan Jarak

Pengujian Alat

Langkah pertama adalah menguji NodeMCU. Pengujian dilakukan dengan menghubungkan mikrokontroler NodeMCU pada USB PC. Apabila LED pada NodeMCU menyala maka alat berfungsi dan bisa digunakan. Setelah melakukan pengecekan hardware, kemudian dilakukan pengujian software mikrokontroler NodeMCU. Pengujian ini dilakukan dengan cara meng-upload program yang sudah dibuat. Software Arduino dapat dinyatakan bekerja secara baik apabila LED berkedip sesuai perintah program yang telah di upload. Selanjutnya melakukan pengujian sensor ultrasonik HC-04 untuk mengukur jarak ketinggian air. Sensor ini dapat mengukur jarak berkisar 2cm hingga 400cm dengan ketelitian mencapai 3 mm.

Pengujian alat secara keseluruhan memerlukan koneksi internet, untuk menghubungkan NodeMCU agar dapat mengirimkan data jarak ketinggian air dari sensor ultrasonik ke aplikasi cayenne dan memberikan notifikasi peringatan ke telegram. Protokol MQTT diperlukan agar pengiriman data terkirim dengan stabil dan tidak ada hambatan. Pengujian ini juga menggunakan LED dan buzzer yang akan menyala pada jarak yang sudah ditentukan untuk menginformasikan keadaan status ketinggian air.

Tabel 2. Rancangan Pengujian

No	Kasus yang Diuji	Hasil yang Diharapkan
1	Pengujian NodeMCU	Dapat berjalan sesuai program.
2	Pengujian sensor ultrasonik	Dapat mendeteksi jarak air yang sudah ditentukan dalam program.
3	Pengujian notifikasi ke <i>telegram</i>	Dapat mengirimkan pesan saat air sudah mencapai status bahaya.
4	Pengujian LED	Lampu akan menyala saat status mencapai batas aman (hijau), waspada (kuning), dan bahaya (merah).
5	Pengujian <i>buzzer</i>	<i>Buzzer</i> akan menyala saat air sudah pada status bahaya.
6	Pengujian <i>platform IoT (cayenne)</i>	Dapat menampilkan jarak ketinggian air yang dideteksi oleh ultrasonik.

HASIL DAN PEMBAHASAN

Hasil Pengujian Akurasi Alat

Pada pengujian akurasi alat ini digunakan modul sensor ultrasonik yang dirangkai dengan NodeMCU/Esp8266 seperti tampak pada gambar 9. Hasil pengujian akurasi alat dapat dilihat pada tabel 3.

Tabel 3. Data Pengukuran Modul Sensor Ultrasonik

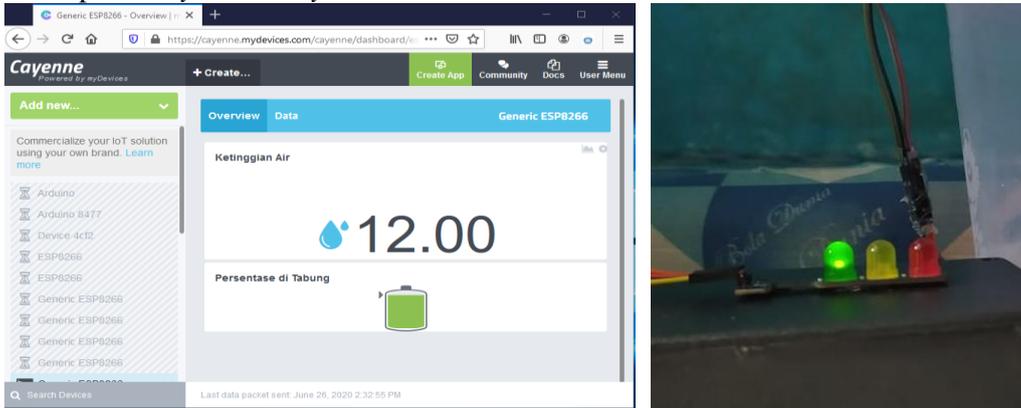
No	Durasi (MS)	Jarak Pengukuran (CM)	Jarak Aktual (CM)	Kesalahan Relatif (%)
1	1749	30,05	30	0,17
2	3507	60,25	60	0,42
3	5243	90,08	90	0,09
4	6998	120,24	120	0,20
5	8748	150,31	150	0,20
6	10494	180,31	180	0,17
7	12218	209,93	210	0,03
8	13970	240,03	240	0,01
9	15713	269,98	270	0,06
10	26190	283,08	283	0,02
% Kesalahan rata-rata				0,125

Pengujian komponen *input* dilakukan untuk memastikan bahwa *input* dapat bekerja dengan baik. Pengujian dilakukan dengan cara memberikan program pada setiap komponen yang digunakan sebagai *input*.

Tabel 4. Hasil Pengujian Alat Sensor Ultrasonik dan NodeMCU

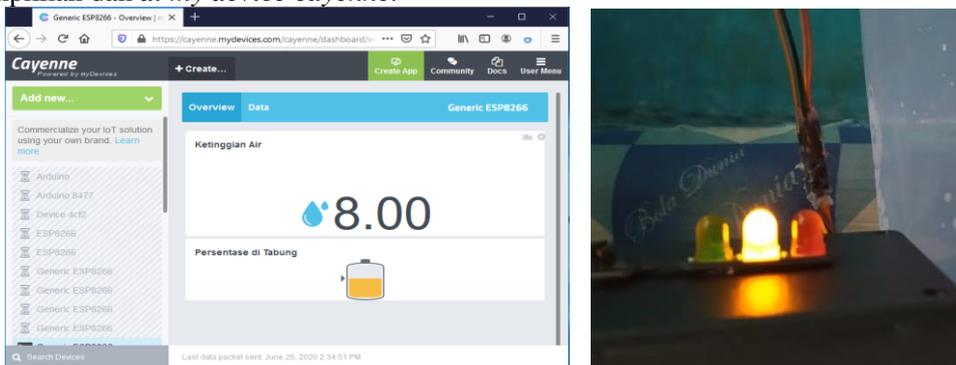
No.	Status Sensor Ultrasonik	NodeMCU	Keterangan
1	Mendeteksi jarak 12-8 cm pada status AMAN	Mengirimkan data	Lampu hijau menyala
2	Mendeteksi jarak 8-4 cm pada status WASPADA	Mengirimkan data	Lampu kuning menyala
3	Mendeteksi jarak 4-0 cm pada status BAHAYA	Mengirimkan data	Lampu merah menyala dan <i>buzzer</i> berbunyi
4	Mendeteksi jarak 4-0 cm pada status BAHAYA	Mengirimkan data ke <i>telegram</i>	Mengirimkan pesan peringatan bahaya banjir ke <i>telegram</i>
5	Mendeteksi setiap jarak yang dideteksi ultrasonik	Mengirimkan data ke <i>cayenne</i>	Menampilkan jarak dan menyimpan data yang dideteksi ultrasonik

Hasil pengujian *output* yang pertama pada jarak 12-8 cm dalam keadaan status AMAN, pada status ini ditandai dengan menyalanya lampu warna hijau dan data jarak ketinggian air akan ditampilkan *my device cayenne*.



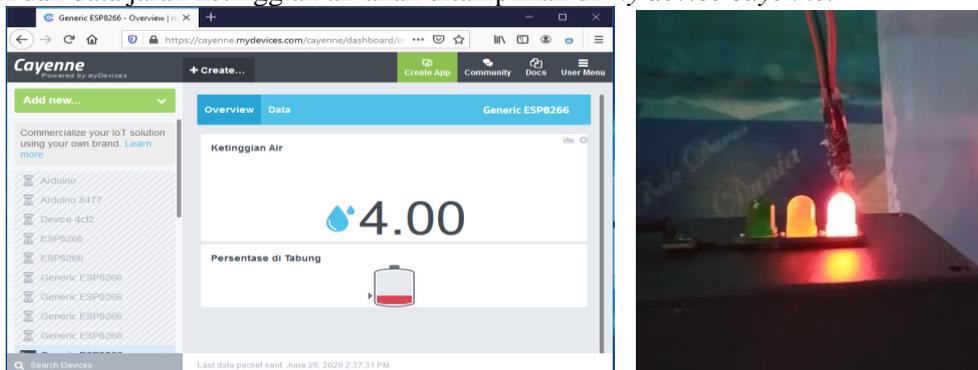
Gambar 9. Tampilan di My Device Cayenne (AMAN) dan LED Hijau

Hasil pengujian *output* yang kedua pada jarak 8-4 cm dalam keadaan status WASPADA, ditandai dengan menyalnya lampu warna kuning dan data jarak ketinggian air akan ditampilkan dan *di my device cayenne*.

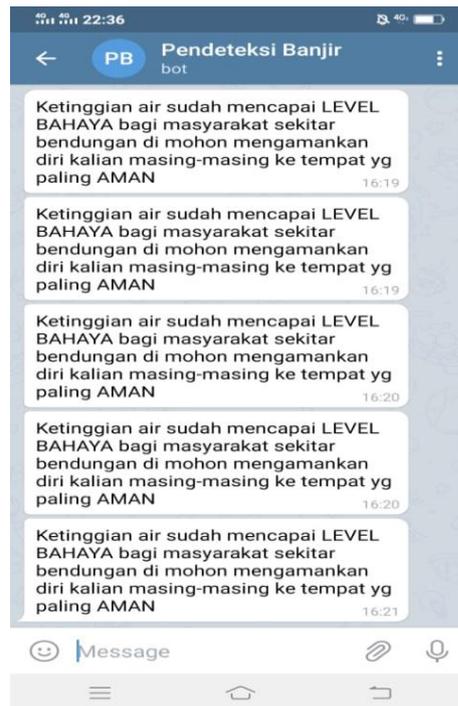


Gambar 10. Tampilan di My Device Cayenne (WASPADA) dan LED Kuning

Hasil pengujian *output* yang ketiga pada jarak 4-0 cm dalam keadaan status BAHAYA, pada status ini di tandai dengan menyalnya lampu warna merah dan diiringi dengan menyalnya *buzzer* selain itu NodeMCU akan mengirimkan notifikasi ke *telegram* berupa pesan peringatan dan data jarak ketinggian air akan ditampilkan di *my device cayenne*.



Gambar 11. Tampilan di My Device Cayenne (BAHAYA) dan LED Merah



Gambar 12. Tampilan Notifikasi Telegram

KESIMPULAN

Berdasarkan hasil penelitian dan pengujian dapat disimpulkan bahwa:

1. Dalam pembuatan *prototype* ini diperlukan beberapa komponen seperti *mikrokontroler* NodeMCU, sensor ultrasonic, LED, *bread board*, *jumper*, *buzzer* dan *black box*.
2. Alat ini mampu memberikan informasi ketinggian air dan ditandai dengan menyalnya lampu LED warna hijau, kuning, dan merah akan menyala sesuai batas yang ditentukan (Aman, Waspada, dan Bahaya) selain itu ketinggian air dapat dimonitoring lewat *platform* IoT dan juga notifikasi pesan *telegram*.
3. Dengan adanya *prototype* sistem pendeteksi banjir menggunakan NodeMCU dan protokol MQTT berbasis IoT yang telah dibuat, masyarakat dapat mengetahui ketinggian air dengan pemanfaatan jaringan *internet* dengan aplikasi *cayenne* secara *realtime*.

SARAN

Dari hasil penelitian dapat diketahui bahwa notifikasi hanya terbatas alarm dan telegram. Oleh karena itu penelitian lanjutannya bisa mengembangkan alat menggunakan notifikasi lain agar penyampaian informasi lebih meluas.

DAFTAR PUSTAKA

- Abilovani, Z.B., Widhi, Y. dan Bakhtiar, F.A. (2018). Implementasi Protokol MQTT untuk Sistem Monitoring Perangkat IoT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* Vol. 2, No. (12), 7521-7527.
- Akhiruddin. (2018). Rancang Bangun Alat Pendeteksi Ketinggian Air Sungai Sebagai Peringatan Dini Banjir Berbasis Arduino Nano. *Journal of Electrical Technology* Vol. 3, No. (3), 174-179.
- Artono, B. dan Putra, R.G. (2017). Penerapan Internet Of Thing (IoT) Untuk kontrol lampu menggunakan Arduino berbasis web. *Jurnal Teknologi Informatika dan Terapan* Vol. 05, No. (01), 9-16.

- Atmoko, R.A. (2019). *Dasar Implementasi Protokol MQTT Menggunakan Phytan dan NodeMCU*. Jakarta: Mokosoft media.
- Danang, Suwardi, dan Ihsan, A.H. (2019). Mitigasi Bencana Banjir dengan Sistem Informasi Monitoring dan Peringatan Dini Bencana menggunakan Microcontroller ArduinoBerbasis IoT. *Jurnal undip* Vol. 40, No. (1), 55-60.
- Dinata, A. (2019). *Fun Coding with MicroPhyton Cara cepat belajar microphyton untuk microcontroller esp8266*. Bandung: Elex Media komputindo.
- Efendi, L. dan Wildian. (2018). Rancang Bangun Sistem Deteksi dan Informasi Lokasi Banjir Berbasis GSM. *Jurnal Fisika Unand* Vol. 7, No. (4), 325- 333.
- Hakim, D.P.A.R., Budijanto, A. dan Widjanarko, B. (2018). Sistem Monitoring Penggunaan Air PDAM Pada Rumah Tangga Menggunakan Mikrokontroler NODEMCU Berbasis Smartphone ANDROID. *Jurnal IPTEK* Vol. 22, No. (2), 9 – 18.
- Hasiholan, C, Primananda, R, dan Amron, K. (2018). Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* Vol. 2, No. (12), 6128-6135.
- Patil, S, et.al. (2019) A Real Time Solution to Flood Monitoring System using IoT and Wireless Sensor Networks. *International Research Journal of Engineering and Technology* Vol. 06, No. (02), 1807-1811.
- Pratama, dan Rizki, P. (2017). Aplikasi Webserver Esp8266 Untuk Pengendali Peralatan Listrik. *Jurnal Informatika Mulawarman* Vol. 17, No. (2), 140-144.
- Rizki, P.P. (2018). Implementasi dan Pengujian Modul ESP8266 dengan Aplikasi Android MQTT-Dash pada Jaringan MQTT. *Jurnal Ilmiah Teknologi Informasi Asia* Vol. 12, No. (2), 157-164.
- Saputra, G.Y., et.al. (2017). Penerapan Protokol Mqtt Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya. *Jurnal Informatika Mulawarman* Vol. 12, No. (2), 70.
- Schwartz, M. (2017). *ESP8266 Internet of Things*. Bandung: Packt Publishing Ltd.
- Siswanto, Nurhadian, T., dan Junaedi, M. (2020). Prototype smart home dengan konsep IoT (Internet of Things) Berbasis NodeMCU dan Telegram. *Jurnal SIMIKA* Vol. 3 No.1, 85-93.
- Umari, U., Anggraini, E., dan Muttaqin, Z.R. (2017). Rancang bangun sistem peringatan dini banjir berbasis sensor ultrasonik dan mikrokontroler sebagai upaya penanggulangan banjir. *Jurnal Meteorologi Klimatologi dan Geofisika* Vol. 4, No. (2), 35-42.