

# Rancang Bangun Prototipe Pemantau Kekeruhan Air dan Pengaturan Pakan Ikan pada Akuarium Menggunakan Nodemcu ESP32

## Prototype Design for Monitoring Water Turbidity and Fish Feed Controlling in Aquariums Using Nodemcu ESP32

Kholidiyah Masykuroh\*<sup>1</sup>, Fikra Titan Syifa<sup>2</sup>, Farhan Ario Pamungkas<sup>3</sup>

<sup>1,2</sup>Prodi S1 Teknik Telekomunikasi, <sup>3</sup>Prodi D3 Teknik Telekomunikasi,  
Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto  
Jalan D.I. Panjaitan No. 128, Purwokerto, Jawa Tengah, Indonesia

\*<sup>1</sup>Penulis Korespondensi: kholidiyah@ittelkom-pwt.ac.id

<sup>2</sup>titan@ittelkom-pwt.ac.id, <sup>3</sup>17201008@ittelkom-pwt.ac.id

Received on 17-12-2022, accepted on 17-01-2023, published on 31-01-2023

### Abstrak

Kemajuan inovasi teknologi saat ini membuat lebih mudah untuk berkomunikasi dari perangkat yang satu ke perangkat lain. Lalulintas informasi antara sumber dan pencari informasi dapat terbantu oleh layanan website. Perkembangan IoT telah mengkombinasikan komunikasi antara perangkat dengan website. Memelihara ikan dalam akuarium merupakan salah satu kegiatan harian masyarakat. Akan tetapi ada permasalahan yang dihadapi yaitu bagaimana memonitor kekeruhan air dan pakan ikan pada akuarium. Kondisi ini berpengaruh pada daur hidup ikan itu sendiri. Oleh karena itu, pada penelitian ini membuat sistem monitoring kekeruhan air dan pakan ikan yang dapat dipantau secara online menggunakan handphone. Prototipe dibuat menggunakan sensor Turbidity untuk memonitor kekeruhan air dan Motor Servo untuk pemberian pakan ikan secara otomatis. Perangkat tersebut dikendalikan oleh mikrokontroler yaitu Nodemcu ESP32. Hasil pengujian yang dilakukan dengan menggunakan sampel acak menunjukkan bahwa sistem bekerja sesuai dengan fungsinya yaitu margin error untuk setiap pengukuran adalah 0,115.

**Kata kunci:** Kekeruhan, NodeMCU, Pakan Ikan, Sensor Turbidity

### Abstract

The advancement in technological innovations today makes it easier to communicate from one device to another. Website services can help information traffic between sources and information seekers. The development of IoT has combined communication between devices and websites. Keeping fish in an aquarium is one of the community's daily activities. However, there are problems faced, namely how to monitor the turbidity of the water and fish feed in the aquarium. This condition affects the life cycle of the fish itself. Therefore, this study created a monitoring system for water turbidity and fish feed that could be monitored online using a mobile phone. The prototype used a Turbidity sensor to monitor water turbidity and a Servo Motor for automatic fish feeding. A microcontroller, namely Nodemcu ESP32, controls the device. The random sample tests show that the system works according to its function. Namely, the margin of error for each measurement is 0.115.

**Keywords:** Fish Feed, NodeMCU, Turbidity, Turbidity Sensor

## I. PENDAHULUAN

Kemajuan inovasi teknologi memudahkan komunikasi antar perangkat. Lalu lintas antara sumber data dan pencari informasi dapat dengan mudah dilakukan melalui layanan website. Perkembangan IoT saat ini memungkinkan untuk mengkombinasikan kedua hal tersebut. Inovasi sistem monitoring berbasis IoT memudahkan penggunaannya untuk memonitor dengan jarak jauh melalui website. Kemudahan ini tentunya membantu berbagai aktivitas manusia [1].

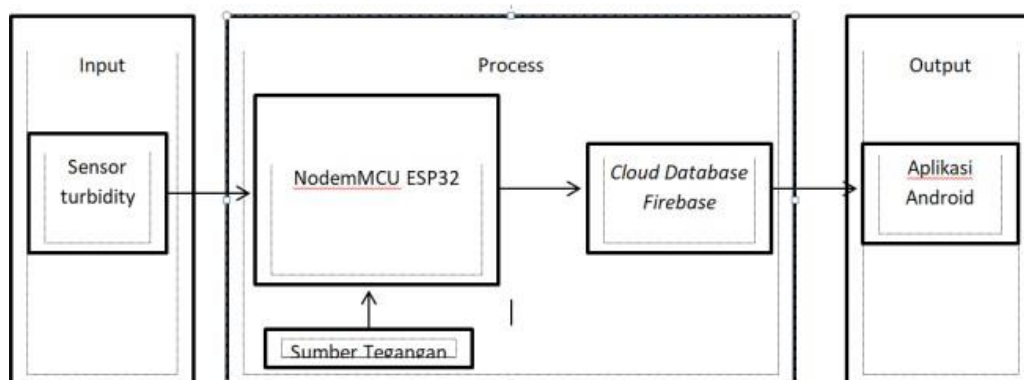
Memelihara ikan dalam akuarium merupakan salah satu aktivitas manusia. Namun untuk menjaga kesehatan ikan peliharaan dalam akuarium tentu menjadi permasalahan. Seperti kualitas air dalam akuarium apakah masih bersih atau sudah kotor dan perlu diganti. Selain itu, pemberian pakan ikan juga menjadi permasalahan. Apabila kedua kegiatan tersebut dilakukan secara manual. Pengembangan instrumen sederhana untuk memonitor akuarium dilengkapi dengan motor servo, alat ini memiliki fungsi untuk membuka dan menutup kompartemen wadah pakan ikan. Instrumen ini juga dilengkapi dengan kamera untuk mengamati apa yang terjadi dalam akuarium. Sistem pemantauan ini dihubungkan dengan web server sebagai antarmuka. Sistem kendali jarak jauh yang diterapkan dengan menggunakan tombol yang deprogram menggunakan JavaScript sebagai tindakan lanjutan dari pengamatan yang dilakukan [2].

Berangkat dari permasalahan tersebut, penelitian ini membahas sistem pemantau kekeruhan air dan pengaturan pakan ikan dalam akuarium menggunakan Nodemcu ESP32. Pemantau kekeruhan air menggunakan sensor Turbidity, sedangkan pengaturan pakan ikan menggunakan Motor Servo.

## II. TINJAUAN PUSTAKA

Pada penelitian ini digunakan bahasa pemrograman Arduino. Kemudian dilakukan pengujian perangkat, baik perangkat monitoring dan pengendalian jarak jauh. Perlu dilakukan pengujian untuk memastikan keduanya dapat bekerja dengan baik. Jika terjadi kesalahan pada saat pengujian maka akan dilakukan perbaikan, apabila semua perangkat dapat bekerja dengan baik maka dilakukan pengujian secara keseluruhan. Terakhir adalah pengumpulan data berdasarkan hasil pengujian perangkat yang dibuat. Berdasarkan data yang diperoleh kemudian dilakukan analisa dan disimpulkan.

Gambar 1 menunjukkan diagram blok implementasi desain sistem pemantauan kekeruhan air di akuarium [3]. Nodemcu ESP32 bertindak sebagai instrumen utama dari prototipe yang dibuat. Nodemcu ESP32 bertugas sebagai mikrokontroler [4] [5], sumber daya untuk menjalankan keseluruhan proses dari prototipe berasal dari catu daya. Sensor kekeruhan kemudian bertindak sebagai input yang mengirimkan data ke Nodemcu. Setelah menerima data, data diproses dan ditransfer ke bagian output. Keluaran sistem adalah aplikasi pada Firebase dan Android. Ketika sensor kekeruhan mendeteksi kurang jernihnya air di dalam akuarium, Nodemcu mengirimkan informasi tersebut ke Google Firebase. Informasi yang diberikan berupa tingkat kekeruhan air yang ada di dalam akuarium [6]. Google Firebase merupakan aplikasi berbasis website. Website ini menyimpan data secara *real time* dan data tersebut dikirim ke aplikasi Android yang telah dibuat menggunakan MIT App Inventor [7].



Gambar 1. Blok Diagram Perancangan Sistem.

### A. NodeMCU ESP32

ESP32 adalah mikrokontroler yang menggantikan ESP8266. Kelebihan dari mikrokontroler ini, ESP32 memiliki modul WiFi. Modul WiFi tersebut dibutuhkan pada saat bekerja pada lingkungan yang berbasis IoT. Gambar 2 menunjukkan *pin out* ESP32. Pin ini dapat digunakan sebagai input atau output untuk menyalakan LCD atau lampu dan menggerakkan motor DC [8].



Gambar 2. NodeMCU ESP32.

### B. Sensor Turbidity

Sebuah sensor kekeruhan yang dapat mendeteksi tingkat kekeruhan air berdasarkan nilai korelasi antara cahaya yang dipantulkan dan cahaya datang. Air memiliki sifat optik yang disebabkan oleh cahaya. Kekeruhan adalah keadaan air yang tidak jernih dan biasanya disebabkan oleh partikel individu (padatan tersuspensi) yang tidak terlihat dengan mata telanjang seperti asap di udara [9].

### C. Motor Servo

Mesin servo adalah mesin listrik yang menggunakan rangka loop tertutup. Bingkai digunakan untuk meningkatkan kecepatan mesin listrik dan untuk mengontrol kecepatan dengan presisi tinggi. Demikian pula, mesin servo biasa digunakan untuk mengubah energi listrik menjadi energi mekanik melalui interaksi dua medan tegangan permanen [10].

### D. Akuarium

Akuarium adalah salah satu bentuk replika kehidupan ikan yang menyerupai lingkungan alam. Akuarium tidak hanya berfungsi sebagai tempat hidup dan berkembang biak ikan, tetapi sering kali digunakan sebagai penghias interior dan dapat pula dimanfaatkan sebagai saluran air. Padahal, baskom kaca ini hanyalah sebuah wadah yang digunakan sebagai tempat hidup dan berkembang biaknya ikan. Selain itu, akuarium menjadi bagian penting bagi kalangan pecinta atau peternak ikan hias [11].

### E. Perangkat Lunak Arduino IDE

*Integrated development environment* (IDE) adalah lingkungan terintegrasi yang disebut iklim. IDE ini dapat digunakan untuk melakukan pembuatan program komputer. Program tersebut bertujuan untuk memperoleh keterampilan yang dimasukkan ke dalam perangkat melalui struktur bahasa pemrograman Arduino. Arduino ini menggunakan bahasa pemrograman yang mirip dengan bahasa pemrograman C. Arduino telah dimodifikasi untuk memudahkan para pemula dalam memprogram untuk pertama kalinya [12].

### F. App Inventor

*Application Inventor* adalah aplikasi *open source* atau dapat dikembangkan lebih lanjut tanpa perlu khawatir mengenai lisensi. Aplikasi ini dibuat oleh Google dan saat ini sedang dikembangkan oleh

*Massachusetts Institute of Technology (MIT). Application Inventor* memungkinkan pelanggan baru memprogram PC mereka untuk membuat aplikasi pemrograman untuk kerangka kerja Android. Dalam membuat App Inventor, Google melakukan survei terkait kemajuan Google berbasis web [13].

### G. Smartphone

Ponsel cerdas adalah ponsel yang lebih kuat, dari tujuan hingga penyorotan dan pemrosesan, karena kerangka kerja yang dapat dikenakan. Kehadiran ponsel ini tentunya siap menawarkan berbagai kemudahan dan kemudahan kepada penggunanya, khususnya para pelajar. Beberapa orang menggunakan ponsel mereka untuk tujuan positif, sementara yang lain menggunakannya untuk tujuan negatif. Ponsel dapat digunakan sebagai alat belajar untuk mempelajari hal-hal baru melalui kesenangan dan berita. Ponsel dapat digunakan sebagai *remote control* yang terhubung ke perangkat pengawasan berbasis Arduino melalui Internet [14].

### H. Google firebase

*Google Firebase* adalah alat Google yang memudahkan pengembang aplikasi untuk mengembangkan ide lebih lanjut. *FirebaseBaaS (Backend as a Service)* adalah jawaban Google untuk menyederhanakan pengembangan pengembang aplikasi seluler. *Firebase* memungkinkan pengembang aplikasi untuk membangun aplikasi mereka tanpa menginvestasikan banyak energi dalam pekerjaan *back-end* [15].

## III. METODE PENELITIAN

### A. Flowchart Alur Sistem

Pada Gambar 3, proses penelitian dimulai dengan tinjauan pustaka, dilanjutkan dengan analisis kebutuhan baik kebutuhan fungsional maupun non-fungsional. Kemudian masuk ke proses desain perangkat keras dan kemudian instal perangkat lunak. Jika terjadi kesalahan, pengujian alat lebih lanjut diulang selama proses desain perangkat, tetapi jika benar, lanjutkan dengan proses akuisisi data.

### B. Skema pengujian sensor Turbidity

Saat menguji sensor kekeruhan, penulis menggunakan beberapa jenis air untuk diukur tingkat kekeruhannya. Hal ini dilakukan untuk membandingkan tingkat kekeruhan air pada beberapa kondisi air. Hasil dari pengukuran menggunakan sensor *turbidity* dibandingkan dengan sensor yang menjadi standar untuk mengetahui akurasi dari sensor *turbidity* yang digunakan.

### C. Pengujian Quality of Service (QoS)

#### 1. Pengujian Throughput

Fitur uji *throughput* untuk mengukur kecepatan data yang dikirimkan dari Nodemcu ESP32 ke Firebase. Pengiriman terjadi pada berbagai jarak yang diujicobakan mulai dari 1-meter sampai dengan 10 meter. Pengujian *throughput* dilakukan secara realtime menggunakan *software* Wireshark yang telah diinstal di laptop.

#### 2. Pengujian Delay

Fungsi dari uji *delay* adalah untuk mencari waktu tunda yang dibutuhkan NodeMCUE SP32 untuk mengirim paket sampai kepada aplikasi penampil yang dibuat pada *smartphone*. Transmisi terjadi pada berbagai jarak yang diujicobakan. Tes *delay* ini juga menggunakan *software* Wireshark. Persamaan 1 di bawah ini menjelaskan cara perhitungan nilai delay dari hasil perbandingan antara waktu antar paket dengan jumlah paket [16].

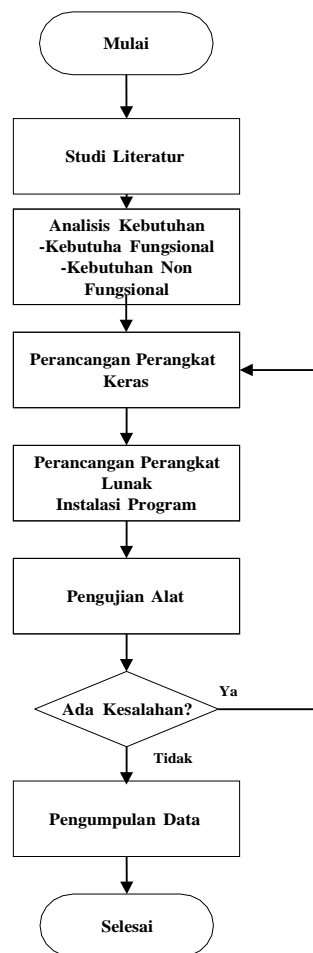
$$\text{delay} = \frac{\text{waktu antar paket}}{\text{jumlah paket}} \quad (1)$$

### 3. Pengujian Packet Loss

Fungsi dari uji *packet loss* adalah untuk mengukur rasio perbandingan antara jumlah paket yang dikirim dan yang diterima. Informasi ini penting untuk mengetahui nilai paket yang hilang pada saat proses pengiriman. Pengiriman paket dilakukan pada berbagai jarak yang diujicobakan mulai dari 1 meter sampai dengan 10 meter. Pengujian *packet loss* dilakukan menggunakan *software* Wireshark yang telah diinstal di laptop. Tabel 2 berikut menjelaskan standar nilai *packet loss* berdasarkan ITU G. 1010 [16].

Tabel 1. Standar Packet Loss Berdasarkan ITU-T G.1010.

Packet Loss (%)	Quality
0	Very Good
5	Good
15	Enough
25	Poor

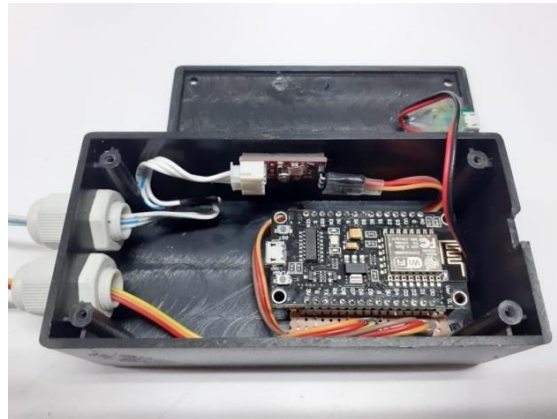


Gambar 3. Flowchart Cara Kerja Prototipe.

## IV. HASIL DAN PEMBAHASAN

Hasil dari pengujian dan pembahasan adalah tahapan-tahapan yang mengikuti proses perancangan dan pembuatan alat pemantau kekeruhan air dan pengaturan pakan ikan. Pengujian yang dilakukan yaitu pengujian sensor kekeruhan dan servomotor. Parameter QoS yang diuji meliputi *packet loss* dan *delay*.

### A. Hasil Perancangan Sistem

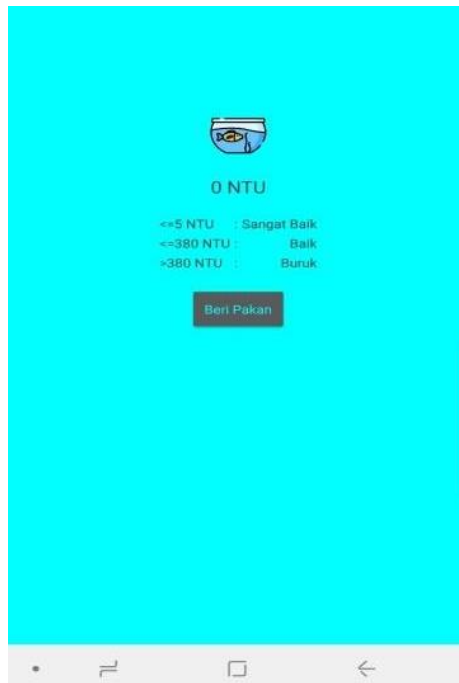


Gambar 4. Hasil Perancangan Hardware.

Gambar 4 menunjukkan hasil perancangan perangkat keras alat pemantau kekeruhan air dan pengaturan pakan ikan. Perangkat keras tersebut memiliki komponen-komponen seperti sensor kekeruhan yang mendeteksi kekeruhan air, servo yang berfungsi sebagai pengatur buka dan tutup wadah pakan ikan, dan NodeMCUESP32 sebagai mikrokontroler.

### B. Hasil Perancangan Perangkat Lunak

Data Firebase dikirim ke aplikasi Android yang dibangun di platform App Inventor.



Gambar 5. Tampilan Screen 1 Aplikasi.

Pada Gambar 5, tampilan aplikasi ditampilkan pada layar 1. Aplikasi ini menampilkan nilai NTU atau kekeruhan pada akuarium. Ada tiga nilai yang harus dilihat: sangat baik, baik dan buruk. Pada Gambar 5, ada tombol pakan ikan yang dapat dioperasikan kapan saja, dan mengkliknya akan membuka atau menutup wadah pakan ikan. Aplikasi ini dapat digunakan secara *real time* pada jaringan yang sesuai.

### C. Hasil Pengujian Sensor Load cell

#### 1. Kalibrasi Turbidity

Sebelum melakukan pengujian sensor kekeruhan, penulis mengkalibrasi setiap elemen dengan kekeruhan air yang berbeda sebanyak empat kali.

Tabel 2. Pengkalibrasian Sensor Turbidity.

No	Jenis sampel air	Nilai ADC (0-1023)	Nilai NTU berdasarkan sertifikat NTU	Hasil Konversi (NTU)	Presentasi eror (%)	Kategori
1	Air Aqua	1023	0.37	0.46	0.24	Jernih
2	Air Teh	933	1.25	1.14	0.08	Kurang Jernih
3	Air Kopi	687	9.09	6.72	0.26	Sangat Keruh
4	Air Susu	0	18.70	19.51	0.04	Sangat Keruh
Rata-rata					0.115	

Tabel 2 menunjukkan hasil kalibrasi sensor kekeruhan air dalam kondisi air yang berbeda-beda seperti air biasa, air teh, air kopi, dan susu. Empat jenis air digunakan sebagai analisa mengenai tingkat kekekeruhan air. Nilai rata-rata presentasi eror sebesar 0,115 atau 11,5% pada saat kalibrasi menunjukkan bahwa sensor kekeruhan dalam kondisi baik.

Gambar 6 menunjukkan program yang digunakan untuk uji kekeruhan.

```

9 // Servo Pakan
10 #include <Servo.h>
11 #define pinPakan D3
12 Servo servopakan;
13 int pakan = 0;
14
15 // Kekeruhan Air
16 #define pinKeruh A0
17 float maxV = 4.2,
18       minV = 2.56;
19 int kekeruhan;
20 String tingkat;
21 int bersih = 25,
22       sedang = 380;
23
24 void wifiFirebase() {
25   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
26   Serial.print("connecting");
27   while (WiFi.status() != WL_CONNECTED) {
28     Serial.print(".");
29     digitalWrite(LED_BUILTIN, HIGH);
30     delay(350);
31     digitalWrite(LED_BUILTIN, LOW);
32     delay(150);
33   }
34   Serial.print("connected: ");
35   Serial.println(WiFi.localIP());
36   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
37 }
    
```

Gambar 6. Program Pengujian Turbidity.

### D. Hasil Pengujian Quality of Service (QoS)

#### 1. Hasil Pengujian Throughput

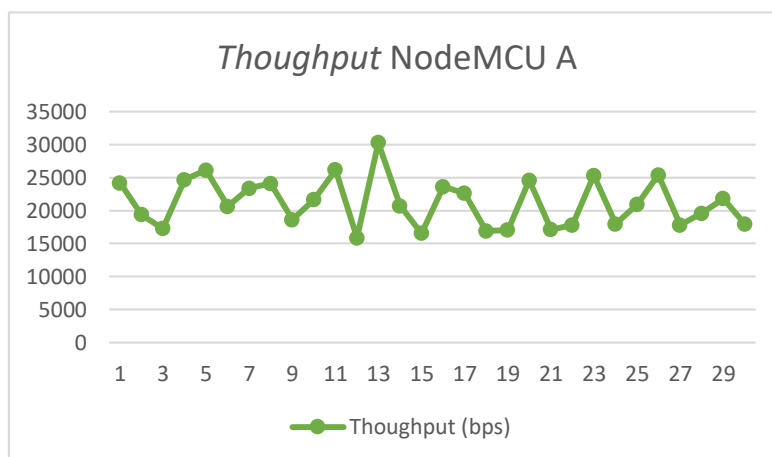
Uji kontinuitas dilakukan pada jarak 1 sampai dengan 10 meter dengan waktu pengujian 1 menit untuk tiap perubahan 1 meternya. Tujuan dari pengujian *throughput* adalah untuk mengukur seberapa banyak data yang diterima dalam hitungan detik. Data hasil pengujian ditunjukkan pada Gambar 7.

Pengujian *throughput*, diuji sebanyak 30 kali dengan volume dan jarak data yang berbeda-beda. Tujuan pengukuran *throughput* pada jarak yang berbeda-beda untuk memberikan nilai *throughput* maksimum dari Nodemcu ESP32.

Pada pengujian *throughput* ini juga ditemukan bahwa nilai *throughput* dipengaruhi banyaknya data dan *delay*, dan semakin besar jumlah data maka semakin besar pula *throughput*nya.

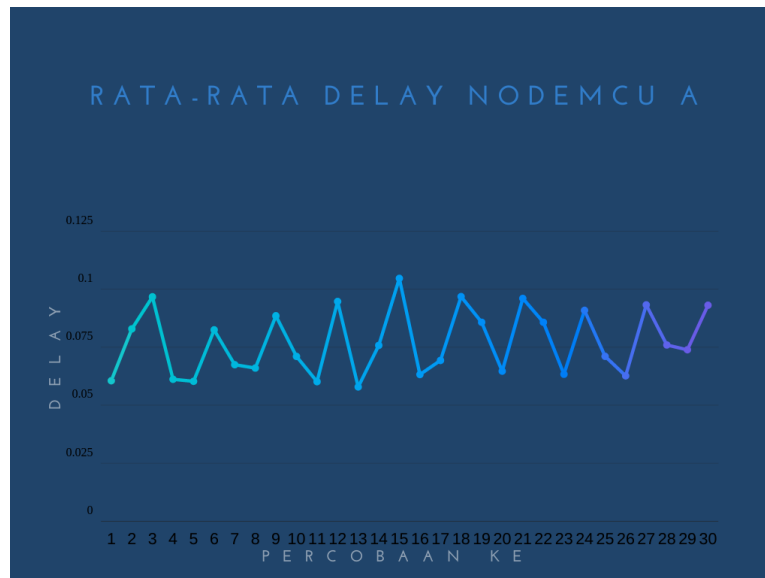
Tabel 3. Hasil Pengujian Packet Loss Pada NodeMCU.

Pengujian	Jarak (meter)	Ukuran data / Paket (Bytes)	Jumlah Data (Bytes)	Paket diterima (a)	Paket ditampilkan (b)	Packet Loss $(\frac{a-b}{a}) \times 100 \%$
1	1	1	91055	498	498	0%
2	1	32	72672	362	362	0%
3	1	64	65372	313	313	0%
4	2	1	93740	497	497	0%
5	2	32	98229	499	499	0%
6	2	64	77375	364	364	0%
7	3	1	88723	450	450	0%
8	3	32	91025	458	458	0%
9	3	64	70143	341	341	0%
10	4	1	81808	426	426	0%
11	4	32	96053	488	488	0%
12	4	64	59779	290	290	0%
13	5	1	114303	522	522	0%
14	5	32	78345	399	399	0%
15	5	64	62804	307	307	0%
16	6	1	89781	481	481	0%
17	6	32	84546	432	432	0%
18	6	64	62804	307	307	0%
19	7	1	64472	353	353	0%
20	7	32	92402	465	465	0%
21	7	64	62700	305	305	0%
22	8	1	66404	357	357	0%
23	8	32	95128	475	475	0%
24	8	64	67888	333	333	0%
25	9	1	78321	422	422	0%
26	9	32	95655	481	481	0%
27	9	64	67295	324	324	0%
28	10	1	74320	400	400	0%
29	10	32	81678	406	406	0%
30	10	64	67562	323	323	0%



Gambar 7. Grafik nilai *throughput*.





Gambar 8. Grafik delay.

## 2. Hasil Pengujian Delay

Uji *delay* berjalan pada jarak 1-10 meter dengan waktu uji 1 menit tiap perubahan 1 meternya. Tujuan dari uji *delay* ini adalah untuk mengetahui waktu tunda yang diperlukan untuk mengirim paket data dari Nodemcu ESP32 ke aplikasi yang telah dibuat pada smartphone sebagai penampil informasi yang dideteksi oleh sensor *Tubidity* dan *motorservo*. Data hasil pengujian ditunjukkan pada Gambar 8.

Uji *delay* ini dilakukan dengan menggunakan data atau paket dengan berbagai jarak dan ukuran paket. Jarak penggunaan adalah 1 hingga 10 meter. Pada setiap jarak, tiga pengukuran dilakukan dengan ukuran paket yang berbeda, yaitu: 1, 32, dan 64 byte. Pengujian ini, penulis menggunakan Nodemcu ESP32 sebagai pusat pengatur sensor dan motorservo, kemudian informasi akan diteruskan ke *Firestore* yang telah dibuat menggunakan MIT APP. Pengujian *delay* dan *packet loss* dilakukan menggunakan laptop yang telah diinstal aplikasi Wireshark. Pengujian dilakukan pada beberapa jarak uji yang berbeda-beda mulai dari 1 meter sampai dengan 10 meter.

## 3. Hasil Pengujian Packet Loss

Pengujian *packet loss* berjalan pada jarak 1 sampai 10 meter dengan waktu pengujian 1 menit tiap perubahan 1 meternya. Tujuan dari pengujian *packet loss* adalah untuk memastikan bahwa jumlah paket yang hilang atau jumlah paket yang dikirim dan jumlah paket yang diterima sudah sesuai. Data hasil pengujian ditunjukkan pada Tabel 3.

Berdasarkan data pada Tabel 3, penulis dapat menyimpulkan bahwa semakin banyak data yang digunakan, semakin banyak paket yang akan ditampilkan. Jumlah paket yang ditangkap hanya bergantung pada jumlah data yang dikirimkan dari Nodemcu ESP32 ke *Firestore*. Hal ini karena jarak yang digunakan dalam pengujian sebenarnya tidak mempengaruhi jumlah paket yang ditangkap dan ditampilkan apabila jumlah paketnya sedikit. Kehilangan paket dapat terjadi apabila jumlah paket yang dikirim mencapai 30 byte upaya dengan jumlah data dan jarak yang berbeda adalah 0%. Artinya Node MCU yang digunakan akan berhasil mengirim semua paket tanpa *packet loss*. Jika paket yang diterima dan paket yang ditampilkan memiliki nilai yang berbeda, ada sebagian data yang dikirim hilang pada saat proses pengiriman dilakukan.

## V. KESIMPULAN

Berdasarkan pengujian dan pembahasan yang dilakukan pada penelitian yang berjudul Rancang Bangun Prototipe Pemantau Kekeruhan Air dan Pengaturan Pakan Ikan pada Akuarium Menggunakan Nodemcu ESP32, maka ditarik kesimpulan sebagai berikut:

Pembuatan alat yang memantau kekeruhan air dan mengontrol pakan ikan IoT, hanya memerlukan tiga komponen perangkat keras: NodeMCU ESP32, sensor kekeruhan, dan servomotor. Kemudian perangkat

lunak yang diperlukan adalah MIT App Inventor untuk mengirimkan informasi yang didapatkan ke aplikasi yang tersedia pada smartphone.

Data dan jarak dalam jumlah besar tidak terlalu memengaruhi penundaan dalam rentang 1 m hingga 10 m. Rata-rata error pada packet loss yang dihasilkan adalah 0%. Nilai throughput maksimum untuk pengujian ke-22 pada jarak 8-meter dan 30 detik adalah 29354.16521 bps, dan nilai throughput minimum untuk pengujian ke-4 pada jarak 2 meter dan 30 detik adalah 9875.199574 bit/detik.

## KONTRIBUSI PENULIS

Penulis pertama bertanggung jawab untuk menyusun artikel yang akan dipublikasikan. Penulis kedua bertanggung jawab untuk mendesain prototipe dan melakukan Analisa hasil penelitian. Dan penulis ketiga bertugas untuk membuat prototipe dan mengumpulkan data hasil percobaan.

## DAFTAR PUSTAKA

- [1] R. A. Wadu, Y. S. B. Ada, and I. U. Panggalo, "Rancang Bangun Sistem Sirkulasi Air Pada Akuarium/Bak Ikan Air Tawar Berdasarkan Kekeuhan Air Secara Otomatis," vol. 1, no. 1, 2017.
- [2] R. Soekarta, D. Yapari, and M. Ackswan, "Rancang Bangun Alat Pemberi Pakan Ikan Otomatis Pada Akuarium Berbasis Arduino Uno," *Insect Inform. Secur. J. Tek. Inform.*, vol. 5, no. 2, p. 16, Mar. 2020, doi: 10.33506/insect.v5i2.1445.
- [3] H. Jemakmun and M. Y. Syamsudin, "Rancang Bangun Akuarium Pintar Berbasis Mikrokontroler," vol. 14, no. 2, 2022.
- [4] M. S. Ramadhan and M. Rivai, "Sistem Kontrol Tingkat Kekeuhan pada Aquarium Menggunakan Arduino Uno," *J. Tek. ITS*, vol. 7, no. 1, pp. 87–91, Mar. 2018, doi: 10.12962/j23373539.v7i1.28499.
- [5] D. Y. Tadeus, K. Azazi, and D. Ariwibowo, "Model Sistem Monitoring pH dan Kekeuhan pada Akuarium Air Tawar berbasis Internet of Things," *METANA*, vol. 15, no. 2, pp. 49–56, Nov. 2019, doi: 10.14710/metana.v15i2.26046.
- [6] D. T. Gandara, *Pemberi Pakan Ikan Otomatis Dengan Pengontrolan Suhu Dan Kejernihan Air Berbasis Arduino*. Universitas Teknologi Yogyakarta, 2018.
- [7] H. Zainul M, A. Faisol, and A. Wahid, "Penerapan Internet Of Things (IoT) Untuk Monitoring Dan Controlling Ph Air Suhu Air Dan Pemberian Pakan Ikan Guppy Pada Aquarium Menggunakan Aplikasi Whatsapp," *JATI J. Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 276–284, Mar. 2022, doi: 10.36040/jati.v6i1.4519.
- [8] E. Systems, "ESP32 Series Datasheet." Espressif Inc, 2019.
- [9] ASAIR, "Datasheet AZDM01." www.aosong.com, 2022.
- [10] S. Electric, "MH3 Servo motor Motor manual." www.schneider-electric.com, 2014.
- [11] A. Hibatullah, *Smart Aquarium Berbasis IoT*. Universitas Muhammadiyah Surakarta, 2019.
- [12] E. E. Prasetyo, "Aplikasi Internet Of Things (IoT) Untuk Pemantauan Dan Pengendalian Beban Listrik Di Ruangan".
- [13] D. Wolber, H. Abelson, E. Spertus, and L. Looney, *App Inventor*, First Edition. United State of America: O'REILLY, 2011.
- [14] I. T. M. Daeng, N. N. Mewengkang, and E. R. Kalesaran, "Penggunaan Smartphone Dalam Menunjang Aktivitas Perkuliahan." e-journal "Acta Diurna," 2017.
- [15] W.-J. Li, C. Yen, Y.-S. Lin, S.-C. Tung, and S. Huang, "JustIoT Internet of Things based on the Firebase real-time database," in *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, Hsinchu, Feb. 2018, pp. 43–47. doi: 10.1109/SMILE.2018.8353979.
- [16] K. Masykuroh, A. D. Ramadhani, and N. Iryani, "Analisis QoS Dan QoE Pada Video Pembelajaran Online di Institut Teknologi Telkom Purwokerto (ITTP)," *Transmisi*, vol. 23, no. 2, pp. 40–47, May 2021, doi: 10.14710/transmisi.23.2.40-47.