

# Perancangan *Water Quality Monitoring* untuk Kolam Ikan Gurami

Imam Syahrizal

Kuliah Kerja Nyata – Pembelajaran Pemberdayaan Masyarakat Periode 2 tahun 2021

Universitas Gadjah Mada

Sleman, Indonesia

<https://github.com/imamsyahrizal>

**Abstrak**—Salah satu mata pencaharian Warga Nanggulan, Kulon Progo, adalah Budidaya Ikan Gurami. Budidaya Ikan Gurami termasuk pekerjaan yang menjanjikan di sisi ekonomi. Namun, budidaya ikan jenis ini memiliki beberapa tantangan dan masalah yang dihadapi. Menurut penjelasan yang diberikan saat acara Penerimaan Mahasiswa KKN-PPM UGM Periode 2 tahun 2021, terdapat dua masalah utama yang dihadapi. Pertama, masalah jamur yang menyerang ikan. Kedua mengenai musang, tetapi masalah mengenai musang ini tidak dijelaskan permasalahan detailnya. Dari dua masalah tersebut, kami lebih berfokus pada masalah jamur. Jamur dapat disebabkan oleh beberapa kondisi salah satunya adalah kualitas air kolam. Kualitas air kolam yang dimaksud adalah suhu air, tingkat keasaman air, dan tingkat kekeruhan air. Parameter-parameter tersebut merupakan besaran yang terukur, sehingga dapat diukur menggunakan sensor. Kita dapat membuat suatu sistem yang dapat mengukur parameter tersebut kemudian akan melakukan aksi ataupun pengambilan keputusan. Pengambilan keputusan ini misalnya adalah ketika nilai yang didapatkan tidak berada di keadaan normal, maka kita bisa melakukan suatu tindakan penanggulangan maupun pencegahan keadaan yang lebih buruk. Pembuatan Sistem *Water Quality Monitoring* ini kami memanfaatkan IoT berbasis Node-RED dan MQTT. Node-RED mampu membuat pengguna (pengelola budidaya) dapat memantau kualitas air dari *smartphone* jika terhubung pada WiFi yang sama dengan sistem.

**Kata kunci**—Gurami, *Internet of Things*, *Monitoring*, *MQTT*, *Node-RED*

## I. LATAR BELAKANG

Budidaya Ikan Gurami merupakan salah satu yang ada di Nanggulan. Salah satu tantangan budidaya Ikan Gurami yang ada di Nanggulan adalah jamur yang menyerang Gurami. Penyakit ini tentu saja merugikan peternak. Ada banyak hal yang mempengaruhi kenapa ikan dapat terserang jamur.

Salah satu penyebab jamur adalah kualitas air kolam menurut Dyah [1] dan Febiola [2]. Kualitas air kolam yang perlu diperhatikan adalah suhu air, tingkat keasaman air, dan kebersihan air. Produktivitas Ikan Gurami dapat terganggu bila berada pada suhu yang dingin serta air yang tidak terlalu asam. Gurami memang dapat hidup dalam kolam yang tidak terlalu bersih, tetapi akan lebih baik dalam kolam yang bersih.

Dari permasalahan tersebut, kami, Tim KKN UGM, mencoba mengembangkan sistem yang dapat memantau ketiga parameter (suhu, tingkat keasaman, dan kebersihan air). Untuk memantau ketiga parameter kami menggunakan tiga buah sensor, sensor suhu DS18B20, sensor pH 4502C, dan sensor kekeruhan *Turbidity Sensor* (TDS). Ketiga bacaan sensor akan dikirimkan ke sebuah server lokal. Untuk melihat hasil pembacaan ketiga parameter tersebut pengguna dapat melihat hasil pembacaan melalui web browser di *smartphone*

ataupun layar LCD yang tersedia. Solusi ini merupakan hasil dari pengembangan dari sistem yang pernah dibuat oleh Sathish [3] Rifai [4], Khairunisa [5], dan Niswar [6]. Dari keempat jurnal tersebut kebanyakan menggunakan *platform* Node-RED dan juga tiga buah sensor juga.

Tujuan dari sistem yang kami buat adalah pengguna dapat memantau nilai suhu, pH, dan tingkat kekeruhan air. Dengan mengetahui besaran parameter tersebut diharapkan jika sedang tidak dalam kondisi normal, maka dapat dilakukan suatu tindakan yang dapat memperbaiki keadaan.

## II. KOMPONEN DAN KONSEP SISTEM

### A. *Komponen Sistem*

#### Raspberry Pi

Raspberry Pi merupakan sebuah komputer papan tunggal atau yang sering disebut dengan *Single Board Computer* (SBC) [7]. Perangkat ini memungkinkan pengguna untuk melakukan berbagai komputasi dan juga belajar pemrograman. Selain itu saat ini perangkat ini juga diharapkan mampu menggantikan fungsi komputer desktop mulai dari menjelajah internet dan memutar video definisi tinggi, hingga membuat *spreadsheet*, pengolahan kata, dan bermain *game* [8].

Kelebihan Raspberry Pi adalah kemampuannya untuk melakukan interaksi dengan dunia luar dan telah banyak digunakan pada berbagai proyek digital [8]. Dalam proyek ini kami menggunakan Raspberry Pi sebagai server lokal yang akan menjadi penghubung antara Node (sensor) dengan pengguna.

Raspberry Pi akan membuat sebuah *dashboard* yang berfungsi untuk menampilkan pembacaan suhu, pH, dan kekeruhan air kolam. Untuk melakukan hal tersebut maka Raspberry akan diinstal sebuah aplikasi bernama Node-RED. Sementara untuk keperluan komunikasinya akan ditangani oleh Mosquitto MQTT.

#### ESP 32

ESP32 merupakan sebuah mikrokontroler berdaya rendah yang terintegrasi dengan WiFi dan *dual-mode* Bluetooth. ESP32 dibuat dan dikembangkan oleh Espressif System, dan dimanufaktur oleh TSMC [9].

Kami memilih ESP32 karena beberapa keunggulannya. Berikut ini merupakan keunggulan-keunggulan dari ESP32 dari situs resminya [10]. Pertama, tangguh, ESP32 didesain tangguh untuk lingkungan industri, dengan suhu kerja di -40 °C sampai dengan 125 °C. Dengan kalibrasi tinggi, ESP bisa beradaptasi dengan lingkungan luar. Kedua, berdaya rendah, ESP32 memiliki konsumsi daya yang rendah karena dirancang untuk kebutuhan perangkat *mobile*, *wearable*

electronic, dan aplikasi IoT. ESP32 juga memiliki fitur-fitur canggih, seperti memiliki beberapa mode *power* dan *dynamic power scaling*. Ketiga, terintegrasi, ESP32 sudah dilengkapi dengan antena, *RF balun*, *power amplifier*, penerima *low-noise*, *filter*, dan modul *power management*. Dengan PCB yang kecil ESP32 memberikan nilai fungsionalitas yang tinggi. Terakhir, *buid-in* WiFi dan bluetooth, ESP32 sudah dilengkapi dengan modul WiFi dan juga bluetooth yang terintegrasi. Sehingga tidak diperlukan lagi modul tambahan.



Gambar 1 ESP32

### Sensor Suhu DS18B20

Sensor suhu yang kami gunakan adalah sensor DS18B20, pemilihan ini berdasarkan referensi dimana sensor jenis ini banyak digunakan dalam proyek *monitoring* kualitas air.

Keunggulan dari sensor ini adalah tingkat keakuratannya, sensor ini memiliki resolusi 12-bit ADC. Dengan 12-bit artinya akurasi sekitar 0.5 °C dalam rentang kerja -10 °C sampai 85 °C. Keunggulan berikutnya adalah sensor ini hanya memerlukan protokol komunikasi 1-kabel (*one-wire*) [11].

Bagian sensor yang terkena air terbuat dari *stainless steel* yang diharapkan tidak mudah berkarat. Sensor ini terdiri dari tiga buah kabel (dua untuk daya dan satu untuk data) yang dibungkus lapisi dengan lapisan tahan air sepanjang 1 meter.

Dilansir dari [ukur.co.id](http://ukur.co.id) [12] ikan gurami akan tumbuh ideal pada temperatur kolam antara 24 °C sampai 28 °C. Bahkan perbedaan suhu antara siang dan malam yang signifikan dapat mengganggu pertumbuhan Ikan Gurami. Menurunnya kadar oksigen dalam kolam juga dapat disebabkan oleh kenaikan suhu air.



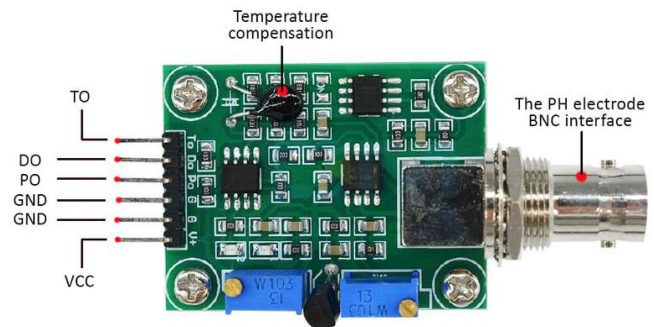
Gambar 2 Sensor Suhu DS18B20 - Sumber Fritzing

### Sensor pH 4502C

Sensor ini akan mengukur tingkat keasaman air kolam. Walaupun terdapat 6 buah pin pada alat, yang akan digunakan hanya ada 3 pin saja. Berdasarkan informasi dari laman [12], Air dengan tingkat keasaman (pH) netral merupakan air yang tepat untuk ikan gurami. Ikan gurami akan berkembang dengan optimal pada pH 6,5 sampai 7,5.

Bagaimana jika nilai yang didapatkan kurang dari 6,5 ? Hal tersebut berarti air kolam bersifat asam. Salah satu cara yang dapat dilakukan untuk menetralkannya adalah dengan mencampurkan kapur ( $\text{CaCO}_3$ ) atau soda kue ke dalam kolam. Sebaliknya jika nilai pH yang didapatkan diatas 7,5 maka air kolam bersifat basa. Dan untuk menurunkan pH-nya dapat dilakukan dengan menambahkan asam fosfor.

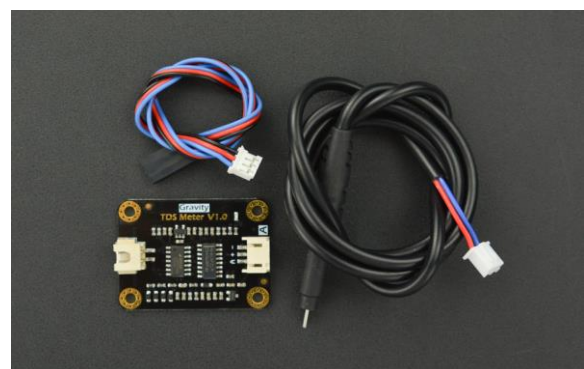
Penambahan-penambahan kapur ataupun asam fosfor harus dilakukan sedikit demi sedikit dan juga disertai dengan mengamati nilai pH yang didapatkan.



Gambar 3 Modul pH meter - Sumber CimpleO

### Sensor Kekeruhan Gravity TDS SEN0244

Berdasarkan laman resmi dari DFROBOT [13] [14] mengenai produk sensor TDS SEN0244 mereka. Sensor Kekeruhan atau *Turbidity Sensor* (TDS) yang kami gunakan produksi dari DFROBOT. Nama sensornya adalah Gravity Analog Meter SEN0244, sensor ini terdiri dari dua bagian yakni Probe dan Modul. *Probe* merupakan bagian yang ada di dalam air, sementara modul adalah bagian yang akan mengirimkan data ke ESP32. Sensor ini mampu mengukur kekeruhan dalam rentang 0 sampai 1000 ppm dengan akurasi kurang lebih 10%. Memerlukan tegangan masukan sebesar 3,3 sampai 5,5 V dengan mengambil 3 sampai 6 mA.



Gambar 4 Sensor TDS - Sumber : DFRobot

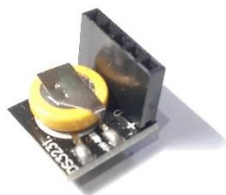
Dan dari penelusuran di laman [15] kekeruhan dapat menurunkan selera makan dan mengganggu pernafasan dari ikan gurami. Menurutnya cara yang bisa dilakukan adalah memasukkan benda berwarna putih sampai 40 cm dari permukaan air. Tingkat kekeruhan tidak mengganggu ikan apabila benda berwarna putih tersebut masih terlihat. Tapi disini kami menambahkan alat ukur kekeruhan sehingga tingkat kekeruhan air dapat terukur.

## Real Time Clock DS3231

Salah satu kekurangan Raspberry Pi adalah tidak adanya modul RTC. Keadaan ini menjadikan Raspberry Pi tidak dapat menghitung waktu dalam keadaan tanpa sumber listrik. Jika Raspberry Pi terhubung ke jaringan internet, hal ini tidak menjadi masalah karena waktu pada Raspberry Pi akan selalu diperbarui saat terhubung dengan internet.

Sistem yang kami rancang telah dibuat sedemikian sehingga tidak memerlukan jaringan internet. Kondisi tersebut karena alasan tertentu, seperti tidak ada jaminan koneksi yang stabil. Hal tersebut menjadikan waktu pada Raspberry Pi akan kacau beberapa saat setelah Raspberry Pi tidak mendapatkan sumber daya listrik. Untuk mencegah masalah tersebut maka kami menambahkan modul RTC yang akan berfungsi sebagai jam pada Raspberry Pi. Modul RTC akan terus bekerja meskipun Raspberry Pi dalam keadaan mati tanpa sumber daya.

Modul RTC akan terhubung dengan Raspberry Pi melalui sambungan I2C. Proses konfigurasi RTC dengan Raspberry Pi sudah dilakukan dan juga sudah melalui proses pengujian. Hal yang mungkin perlu dilakukan beberapa bulan sekali adalah memastikan waktu pada RTC tetap tepat. Jika tidak sesuai maka dapat diatur melalui Raspberry Pi yang dihubungkan dengan jaringan internet. Jika memerlukan tutorial dapat melihat *logbook* mengenai pembuatan sistem ini, disana sudah ada langkah-langkahnya.

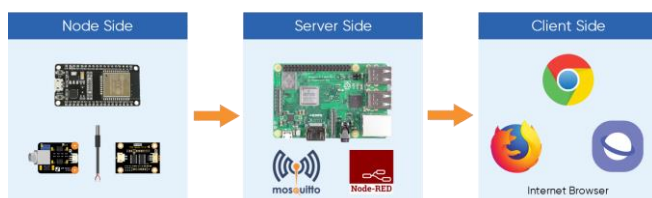


Gambar 5 Real Time Clock DS3231

## B. Konsep Sistem System Overview

Sistem yang kami buat secara sederhana dapat digambarkan dengan diagram berikut ini

### Water Quality Monitoring



Gambar 6 Diagram Sistem

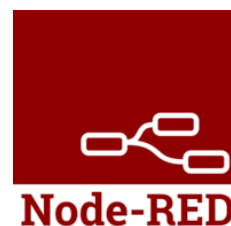
Secara umum sistem tersebut terbagi menjadi tiga bagian, Server side, Client side, dan juga Sensing Node Side.

## Server Side

Pada bagian server, komponen utama yang kami pakai adalah Raspberry Pi. Di dalam Raspberry Pi tersebut sudah terpasang Node-RED dan juga Mosquitto MQTT Broker.

### Node-RED

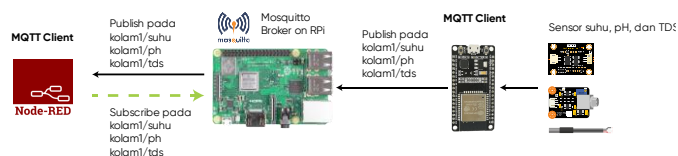
Node-RED merupakan program *open source* dan berbasis *flow* (bukan kode), biasa digunakan untuk melakukan hubungan IoT, APIs, dan *services*. Dengan Node-RED yang menggunakan pendekatan pemrograman visual, memungkinkan pengembang hanya cukup menghubungkan *node-node* yang tersedia. Kode dapat ditulis dalam bahasa HTML dan JavaScript. Di dalam Node-Red dikenal blok-blok dengan sebutan *node*. Masing-masing *node* ini memiliki fungsinya masing-masing [6].



Gambar 7 Icon Node-RED

### Mosquitto MQTT Broker

Sebelum membahas apa itu Mosquitto MQTT Broker, terlebih dahulu akan dibahas mengenai apa itu MQTT. Menurut Rui [16] MQTT merupakan protokol pengiriman pesan yang simpel dan didesain untuk perangkat dengan *bandwidth* yang rendah. Saat ini MQTT merupakan solusi yang terbaik dalam *Internet of Things*. MQTT bisa memfasilitasi pengiriman perintah ke *aktuator*, membaca dan mempublikasi data dari sensor dan banyak hal lainnya.



Gambar 8 Contoh cara kerja MQTT

*Sensing node* lebih tepatnya ESP32 melakukan *publish* data suhu pada topik **kolam1/suhu**, pH pada topik **kolam1/ph**, dan kekeruhan pada topik **kolam1/tds**. Di sisi server Node-RED yang terinstal di Raspberry Pi B+ melakukan *subscribe* dengan ketiga topik tersebut. Jadi, nilai suhu, pH, dan kekeruhan dapat ditampilkan dalam bentuk grafik.

### Publish dan Subscribe

Konsep penting pertama adalah *Publish* dan *Subscribe*. Suatu perangkat dapat melakukan *publish* sebuah pesan dalam topik tertentu, atau juga dapat melakukan *subscribe* ke topik tertentu.



Gambar 9 Publish dan Subscribe



Sebagai contoh Perangkat 1 melakukan *publish* pada topik tertentu. Kemudian Perangkat 2 melakukan *subscribe* pada topik yang sama dengan Perangkat 1. Sehingga Perangkat 2 akan menerima pesan yang di *publish* oleh Perangkat 1.

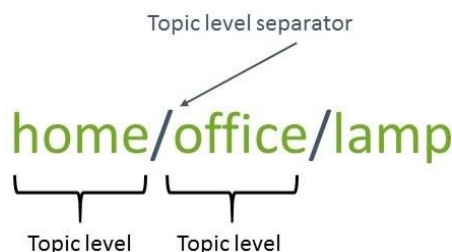
### Messages

Sama seperti terjemahannya, pesan, *message* berarti sebuah informasi yang akan ditukarkan di antara dua atau lebih perangkat. Pesan tersebut bisa berupa perintah ataupun sebuah data. Pada sistem ini pesan yang akan ditukarkan adalah data bacaan sensor.

### Topics

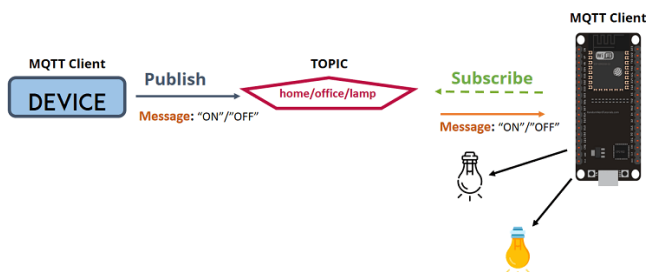
Konsep yang selanjutnya penting adalah topics, Topics merupakan cara untuk melakukan registrasi pesan seperti pesan apa yang ingin diterima dan dipublikasikan. Topik biasanya direpresentasikan dengan *strings* yang dipisahkan dengan tanda garis miring “/”. Setiap garis miring menandakan level dari topik.

Berikut ini contoh ilustrasi topik untuk sebuah lampu yang ada di rumah lebih tepatnya di ruang kerja



Gambar 10 Topik MQTT

Kita akan menyalakan lampu tersebut menggunakan MQTT, berikut ini merupakan skenarionya



Gambar 11 Contoh MQTT – Sumber Rui Santos

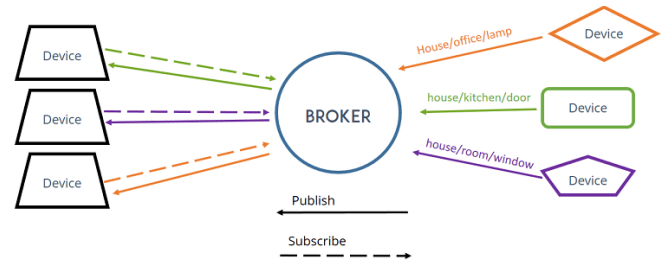
Terdapat sebuah perangkat yang mempublikasikan pesan “on” dan “off” pada topik **home/office/lamp**. Di sisi lain juga terdapat perangkat yang mengontrol lampu dalam hal ini adalah *microcontroller* ESP32. ESP32 tersebut melakukan *subscribe* pada topik **home/office/lamp**.

Maka ketika ada pesan baru yang di publikasi pada topik tersebut (**home/office/lamp**), ESP32 akan menerima pesan “on” atau “off” dan akan menghidupkan atau mematikan lampu sesuai dengan perintah yang diterima.

### Broker

Broker merupakan suatu *tools* yang bertanggung jawab untuk menerima semua *message* (pesan), memfilter pesan, dan memutuskan siapa yang akan menerima pesan tersebut,

kemudian akan dilakukan publikasi pesan ke semua klien yang ter-*subscribe*, berikut ini contoh diagram *subscriber*



Gambar 12 Broker - Sumber : Rui Santos

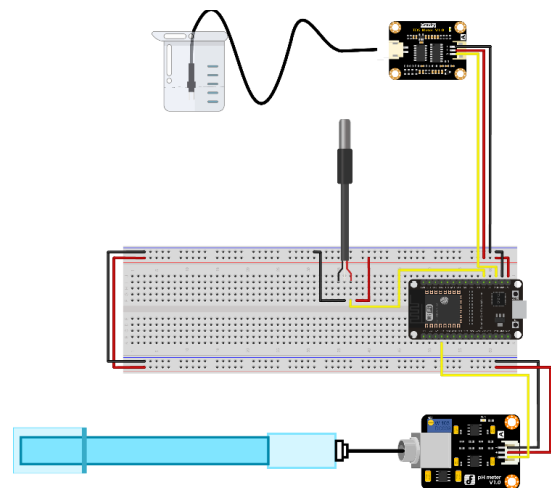
Salah satu broker yang populer adalah Mosquitto MQTT, dan ini adalah broker yang kami pakai.



Gambar 13 Logo MosquittoMQTT Broker

### Sensing Node Side

*Sensing Node* adalah bagian dari sistem yang akan mengambil data. Data yang diambil adalah suhu, tingkat pH, dan tingkat kekeruhan air. Komponen utama bagian ini adalah ESP32 yang berfungsi sebagai *microcontroller*. Komponen lainnya adalah sensor suhu DS18B20, pH 4502C, dan TDS SEN0244. Komponen-komponen tersebut dirangkai seperti rangkaian berikut



Gambar 14 Sensing Node Diagram

Untuk program .ino dapat ditemukan pada *repository* github, link *repository* akan diberikan di bagian selanjutnya.

### III. DISKUSI SISTEM WATER MONITORING

#### Log Pembuatan Sistem

Untuk beberapa proses penting seperti memasang Mosquitto, Node-RED, dan hal yang dianggap penting terdapat langkah-langkah step by step yang dilakukan sebagai dokumentasi jika diperlukan pada Logbook Perancangan Sistem. Pada dasarnya tulisan tersebut bukanlah tutorial melainkan hanya catatan proses pengembangan sistem.

#### Konfigurasi Raspberry Pi

Akun *root* Raspberry Pi sudah bukan versi *default* bawaan. Saat ini *username* : **pi** dengan *password* : **raspberrypi** *username* dan *password* ini diperlukan ketika akan mengakses konfigurasi Raspberry Pi ataupun saat menggunakan *remote* akses.

#### Konfigurasi WiFi Router

Router yang kami berikan adalah Mercusys MW302R. Sudah ada beberapa konfigurasi yang diatur pada Router tersebut. Pertama, mode saat ini adalah mode router. Selanjutnya, SSID : **YO123** dengan Password : **kknppmugm**. Terakhir, DHCP untuk reservasi alamat *IP Static* Raspberry Pi di **192.168.1.100**. Jika ingin membuka pengaturan Router dapat dibuka pada link <http://mwlogin.net>. Saat log-in ke pengaturan WiFi gunakan *username* : **Admin123** dengan *Local password* : **Admin123**.

#### Dokumentasi Kode dan Github

Semua kode dan hal yang berkaitan dengan perancangan sistem ini akan didokumentasikan dan disimpan pada *repository* Github <https://github.com/imamsyahrizal/KKN-Nanggulan>, *repository* tersebut dapat digunakan sebagai referensi pengembangan sistem selanjutnya

### IV. SARAN PENGEMBANGAN

Tahap yang sudah dilakukan sampai saat ini adalah pembuatan *prototype* sistem. Sistem yang sudah dibuat pada dasarnya sudah bisa langsung digunakan, tapi masih belum optimal di bagian tertentu.

Pertama, *dashboard* yang akan menampilkan parameter kualitas air kolam masih berupa *dashboard* standar dari Node-RED. *Dashboard* ini bisa dibuat sesuai dengan keinginan sendiri. Saat pembuatan sistem ini tidak sampai membuat *custome dashboard* dikarenakan waktu yang terbatas sehingga lebih mengutamakan selesainya fungsi-fungsi penting dan juga dokumentasi sistem (laporan, poster, dan *repository* kode). Pembuatan *dashboard* ini bisa menggunakan grafana.

Kedua, Raspberry Pi adalah *Single Board Computer*, yang memiliki makna lain “komputer mini”. Sebuah komputer umumnya terdiri dari CPU (dalam hal ini Raspberry Pi) dan periferal lainnya seperti *mouse*, *keyboard*, dan *monitor*. Untuk saat ini yang sudah ada hanyalah monitor LCD berukuran 3,5 inch sebagai pengganti monitor. Selain itu kami juga jarang menggunakan *mouse* dan *keyboard* sebagai masukan karena kami mengendalikan Raspberry Pi dari PuTTY atau VNC. Mengendalikan dengan cara tersebut bisa dilakukan secara *headless* (tanpa masukan apa pun).

Ketiga, sampai saat ini kami hanya membuat sistem yang dapat memonitoring sebuah kolam saja. Untuk bisa memonitoring lebih dari satu kolam hanya perlu menambah *Sensing Node* (ESP32 + sensor) dan mengubah sedikit kode pada Node-RED dan ESP. Sehingga kami menyarankan agar membuat lagi *sensing node* agar sistem ini bisa memonitoring lebih banyak kolam ikan.

### V. DAFTAR PUSTAKA

- [1] D. S. Fitri, “Penyakit Jamur Pada Gurame,” GDMORGANIC, 24 Mei 2021. [Online]. Available: <https://gdmorganic.com/cara-mengatasi-ikan-gurame-yang-berjamur/>. [Diakses Juli 2021].
- [2] F. N. S., “Saprologniasis, Penyakit Jamur yang Sering Menyerang Ikan Air Tawar,” Desa Sendang, 9 Agustus 2020. [Online]. Available: <http://sendang-wonogiri.desa.id/2020/08/09/saprologniasis-penyakit-jamur-yang-sering-meny Serang-ikan-air-tawar/>. [Diakses Juli 2021].
- [3] S. Pasika dan S. T. Gandla, “Smart water quality monitoring system with cost-effective using IoT,” *Heliyon*, vol. 6, no. 7, 2020.
- [4] A. Rifa'i, M. U. H. Rasyid dan A. I. Gunawan, “Sistem Pemantauan dan Kontrol Otomatis Kualitas Air Berbasis IoT Menggunakan Platform Node-RED untuk Budidaya Udang,” *Jurnal Teknologi Terapan*, vol. 7, no. 1, pp. 19-26, 2021.
- [5] K. M. dan Y. Irawan, “Smart Aquarium Design Using Raspberry Pi and Android Based,” *Jurnal of Robotics and Control*, vol. 2, no. 5, pp. 368-372, 2021.
- [6] M. Niswar, S. Wainalang, A. A. Ilham, Z. Zainuddin, Y. Fujaya, Z. Muslimin, A. W. Paundu, S. Kashiara dan D. Fall, “IoT-based Water Quality Monitoring System for”.
- [7] “Raspberry Pi,” Wikipedia, 20 Desember 2019. [Online]. Available: [https://id.wikipedia.org/wiki/Raspberry\\_Pi](https://id.wikipedia.org/wiki/Raspberry_Pi). [Diakses Juli 2021].
- [8] “What is a Raspberry Pi?,” Raspberry Pi, [Online]. Available: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Diakses Juli 2021].
- [9] “ESP32,” Wikipedia, 22 Juli 2021. [Online]. Available: <https://en.wikipedia.org/wiki/ESP32>. [Diakses Juli 2021].
- [10] “ESP32,” Espressif, [Online]. Available: <https://www.espressif.com/en/products/socs/esp32>. [Diakses Juli 2021].
- [11] K. Wardana, “[TUTORIAL] Menggunakan Sensor Suhu DS18B20 pada Arduino,” Tutorialkaren, 10 Juni 2016. [Online]. Available: <https://tutorkeren.com/artikel/tutorial-menggunakan-sensor-suhu-ds18b20-pada-arduino.htm>. [Diakses Juli 2021].
- [12] “Air Untuk Ikan Gurame,” Solusi Pengukuran, [Online]. Available: <https://ukur.co.id/air-untuk-ikan-gurame-2/>. [Diakses Juli 2021].
- [13] “Gravity Analog TDS Sensor Meter For Arduino SKU SEN0244,” DFROBOT, [Online]. Available: [https://wiki.dfrobot.com/Gravity\\_Analog\\_TDS\\_Sensor\\_Meter\\_For\\_Arduino\\_SKU\\_SEN0244#top](https://wiki.dfrobot.com/Gravity_Analog_TDS_Sensor_Meter_For_Arduino_SKU_SEN0244#top). [Diakses Juli 2021].
- [14] “Gravity: Analog TDS Sensor/Meter for Arduino,” DFROBOT, [Online]. Available: <https://www.dfrobot.com/product-1662.html>. [Diakses Juli 2021].
- [15] “6 Tips Menentukan Kualitas Air Kolam Budidaya Ikan Gurami,” Budidaya Ikan Gurami, [Online]. Available: <http://belajarbudidayagurami.blogspot.com/2014/04/6-tips-menentukan-kualitas-air-kolam-untuk-budidaya-ikan-gurami.html>. [Diakses Juli 2021].
- [16] R. Santos, “What is MQTT and How It Works,” Random Nerd Tutorials, [Online]. Available: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>. [Diakses Juli 2021].