

# AQUARIUM WATER TURBIDITY METER USING IOT CLOUD COMPUTING SERVICES

## PENGUKUR KEKERUHAN AIR AKUARIUM MENGGUNAKAN LAYANAN KOMPUTASI AWAN IOT

Falentino Gilbert Assa, Sherwin Reinaldo U Aldo Sompie

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

E-mail: 14021106045@student.unsrat.ac.id, aldo@unsrat.ac.id

**Abstract** — Aquarium water turbidity gauges using IoT cloud computing services are created to help owners or businesses of ornamental fish maintain the cleanliness of water in the aquarium. Aquarium water turnover is mandatory because aquarium water can be contaminated from food waste, urine, and fish feces. In addition, reduced minerals and water pH can badly affect the strength and health of fish. Therefore, without regular water turnover, fish will not be able to live healthy and will soon die in vain. The rapid development of science and technology in the field of electronics and instrumentation has made it possible to create measuring instruments that work digitally and can interact with each other. This can help ornamental fish owners to make it easier to do aquarium maintenance because it no longer needs to be done manually and can be done anywhere. This device is based on nodeMCU esp8266 microcontroller and uses Analog turbidity sensor. The output of this tool is processed by the Arduino program and displayed on Thingspeak.

**Key words**— IoT, turbidity, esp8266, NodeMCU, Analog Turbidity Sensor, Arduino, Thingspeak.

**Abstrak** — Alat pengukur kekeruhan air akuarium menggunakan layanan komputasi awan IoT dibuat untuk membantu pemilik ataupun pelaku usaha ikan hias menjaga kebersihan air dalam akuarium. Pergantian air akuarium adalah hal yang wajib dilakukan karena air akuarium dapat terkontaminasi dari sisa makanan, urin, dan tinja ikan. Selain itu berkurangnya mineral dan pH air dapat berdampak buruk pada kekuatan dan kesehatan ikan. Oleh karena itu, tanpa pergantian air yang rutin, ikan tidak akan bisa hidup sehat dan dengan segera akan mati sia-sia. Perkembangan ilmu pengetahuan dan teknologi yang demikian pesat di bidang elektronika dan instrumentasi telah memungkinkan diciptakannya alat-alat ukur yang bekerja secara digital dan dapat saling berinteraksi satu sama lain. Hal ini dapat menolong para pemilik ikan hias agar lebih mudah melakukan perawatan akuarium karena tidak perlu lagi dilakukan secara manual dan bisa dilakukan dimana saja. Alat ukur ini berbasis mikrokontroler esp8266 NodeMCU dan menggunakan Analog urbidity Sensor. Keluaran dari alat ini diolah oleh program Arduino dan ditampilkan pada Thingspeak.

**Kata kunci** — IoT, kekeruhan, esp8266, NodeMCU, Analog Turbidity Sensor, Arduino, Thingspeak.

### I. PENDAHULUAN

Keberadaan air bersih adalah hal yang vital bagi pemilik ikan hias. Terlepas dari jenis ikan yang dimiliki, pergantian air akuarium adalah hal yang wajib dilakukan. Alasannya adalah

karena air akuarium dapat terkontaminasi dari sisa makanan, urin, dan tinja ikan. Selain sampah dan kotoran, adanya produk sampingan limbah yang tak terlihat mata menumpuk di akuarium dalam bentuk nitrat dan fosfat dapat membuat ikan stress dan rentan terhadap penyakit. Kotoran bukanlah satu-satunya alasan mengapa air perlu diganti. Berbagai elemen mineral dalam air penting bagi kesehatan ikan serta stabilitas kimiawi air. Seiring waktu, mineral-mineral itu akan habis atau tersaring keluar. Jika tidak diganti, pH air akan turun. Lebih jauh lagi, kurangnya kadar mineral akan berdampak buruk pada kekuatan dan kesehatan ikan. Oleh karena itu, tanpa pergantian air yang rutin, ikan tidak akan bisa hidup sehat dan dengan segera akan mati sia-sia.

Pada masa kini masih banyak orang yang melakukan pergantian air akuarium secara manual, ada juga orang-orang yang karena kesibukan sehari-hari lupa atau tidak memiliki waktu untuk mengganti air akuarium mereka. Perkembangan ilmu pengetahuan dan teknologi yang demikian pesat di bidang elektronika dan instrumentasi telah memungkinkan diciptakannya alat-alat ukur yang bekerja secara digital dan dapat saling berinteraksi satu sama lain. Hal ini dapat menolong para pemilik ikan hias agar lebih mudah melakukan perawatan akuarium karena tidak perlu lagi dilakukan secara manual dan bisa dilakukan dimana saja.

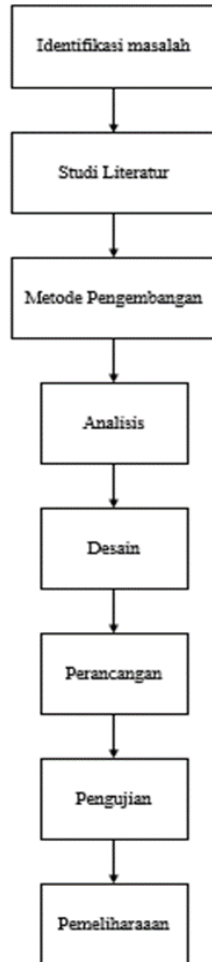
Berdasarkan uraian diatas maka, penulis akan merancang dan membangun suatu alat yang akan mengukur tingkat kekeruhan zat cair ini menggunakan Analog Turbidity Sensor, dimana sensor ini dapat mendeteksi tingkat kekeruhan air dengan cara mengukur jumlah intensitas cahaya yang tersebar pada cairan. Sistem yang dibangun ini berbasis pada NodeMCU.

### II. METODE

#### A. Waktu dan tempat

Penelitian ini dilaksanakan di Desa Mapanget, Kecamatan Talawaan, Minahasa Utara dan lama penelitian ini akan dilaksanakan dari bulan Oktober 2020 sampai dengan bulan Juni 2021.

## B. Kerangka pikir



Gambar 1. Kerangka pikir

## C. Metode pengembangan

Metode yang akan digunakan untuk mengembangkan perangkat pada penelitian ini adalah metode waterfall. Menurut Pressman (2012) metode waterfall adalah metode klasik yang bersifat sistematis, berurutan dalam membangun software. Tahapan – tahapan pada model ini adalah analisis, perancangan, implementasi, pengujian dan pemeliharaan.

### 1. Analisis

Langkah pertama yang akan dilakukan oleh peneliti pada metode ini adalah dengan menganalisis dan mengumpulkan kebutuhan yang diperlukan oleh sistem yang akan dibangun. Kebutuhan sistem seperti perangkat keras dan perangkat lunak yang diperlukan sistem akan diidentifikasi pada tahap ini. Langkah ini juga diperlukan agar dapat ditentukan fitur apa saja yang perlu ada pada sistem.

### 2. Perancangan

Pada tahap ini peneliti akan menterjemahkan hasil dari analisis kebutuhan dan syarat pada tahap sebelumnya kedalam bentuk perancangan yang akan menjadi acuan pada tahap implementasi. Pada tahap ini akan dibuat use case, activity diagram, dan desain perangkat keras.



Gambar 2. Use case diagram

### 3. Implementasi

Pada tahap ini desain dan perancangan yang telah dibuat pada tahap sebelumnya akan diimplementasikan. Implementasi diawali dengan memasang rangkaian perangkat keras mengikuti desain perangkat keras yang telah dibuat sebelumnya. Komponen utama perangkat keras adalah NodeMCU, Turbidity Sensor, relay, motor servo, dan pompa bertegangan 5v. Setelah perangkat keras sudah terpasang, selanjutnya akan diinstal perangkat lunak Arduino IDE. Alat yang akan dibuat berdasarkan use case dan activity diagram yang telah dirancang pada tahap sebelumnya.

### 4. Pengujian

Sistem yang telah diimplementasikan pada tahap sebelumnya akan diuji. Pada penelitian ini alat yang telah dibuat akan diuji langsung pada wadah akuarium serta dihubungkan dengan berbagai perangkat keras yang telah disiapkan. Pengujian akan dilakukan pada sensor, motor servo, pompa, dan kemudian pengujian akhir berupa pengujian ketiga alat secara bersamaan.

### 5. Pemeliharaan

Pada tahap ini peneliti akan memperbaiki kesalahan – kesalahan pada sistem yang baru ditemukan pada saat sistem dijalankan dan belum pernah ditemukan pada tahap sebelumnya.

## D. Desain sistem

Pada tahap ini peneliti akan menggambarkan bagaimana pengembangan sistem yang akan peneliti lakukan dengan menggunakan metode diagram UML.

### 1. Use case diagram

Use Case Diagram akan menggambarkan bagaimana sistem yang akan dibuat peneliti menunjukkan interaksi antara pengguna sistem.

### 2. Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam aplikasi yang sedang dikembangkan, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.

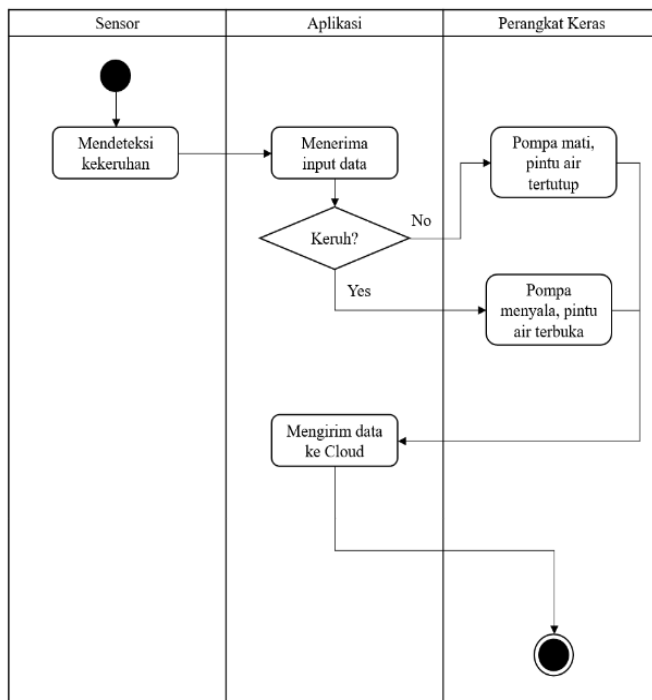
Pada gambar 3 diketahui bahwa proses pada aplikasi dimulai dengan sensor yang akan mendeteksi nilai kekeruhan air dalam wadah, hasil bacaan sensor kemudian diteruskan menuju aplikasi. Jika nilai bacaan menunjukkan kekeruhan pada air maka pompa akan menyala dan pintu air akan dibuka. Sebaliknya, jika nilai bacaan tidak menunjukkan kekeruhan pada air maka pompa akan mati dan into air akan ditutup. Hasil bacaan dari aplikasi kemudian akan diteruskan ke Cloud agar bisa dilihat secara langsung oleh publik.

#### E. Perancangan perangkat keras

Untuk membantu peneliti agar memiliki gambaran bagaimana perangkat keras pada sistem akan diimplementasikan nantinya maka diperlukan desain skema perangkat keras. Desain ini nantinya akan menjadi acuan agar perangkat keras yang dirancang sesuai dan tidak melenceng dari tujuan awal. Untuk lebih jelasnya, gambar desain skema perangkat keras dapat dilihat pada gambar 4.

Desain Skema menggunakan NodeMCU v3 esp8266 Lolin, Analog Turbidity Sensor, relay, pompa mini 5V, motor servo, dan Adaptor 5V. Setiap perangkat memiliki peran tersendiri dan akan dihubungkan sesuai yang terlihat pada gambar 4.

NodeMCU merupakan perangkat elektronika utama karena aplikasi yang akan dibangun pada penelitian dibuat dan berjalan pada mini komputer ini. NodeMCU juga akan mengontrol dan menjadi pengalir energi listrik bagi perangkat – perangkat elektronika yang lainnya.

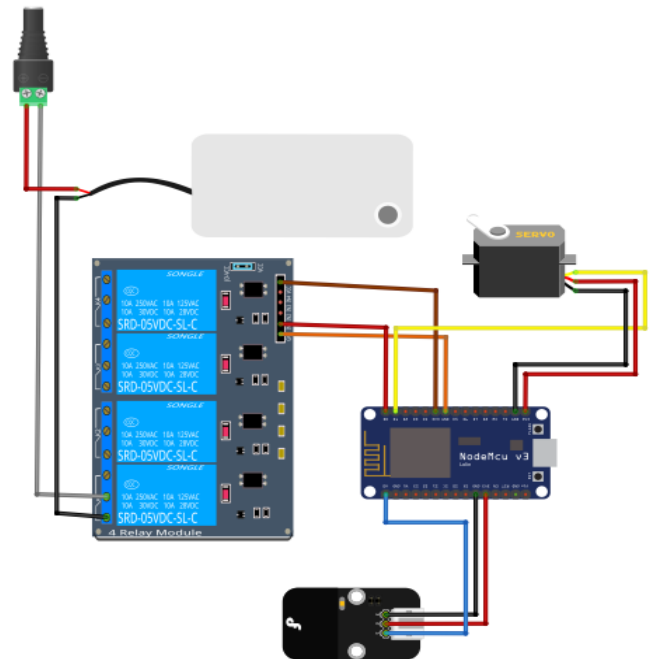


Gambar 3. Activity diagram aplikasi

Analog Turbidity Sensor merupakan perangkat input untuk aplikasi. Fungsinya adalah untuk mendeteksi partikel tersuspensi dalam air dengan cara mengukur transmitansi dan hamburan cahaya yang berbanding lurus dengan kadar Total Suspended Solids (TSS). Perangkat ini terhubung ke NodeMCU melalui pin power 3V, pin ground, serta kabel signal yang dhiubungkan dengan pin A0.

Relay berfungsi sebagai saklar elektrik yang mana memutus dan menghubungkan aliran listrik pada sebuah rangkaian dengan kontrol berupa tegangan yang masuk pada bagian coilnya. Relay ini memiliki 4 channel yang bisa digunakan dan terhubung ke NodeMCU melalui 3 pin, kabel positif yang dihubungkan ke NodeMCU melalui pin power 3V, kabel signal yang dihubungkan dengan pin D0, dan 1 kabel yang terhubung dengan pin ground.

Motor servo dan pompa air merupakan perangkat output. Motor servo berfungsi untuk mengangkat pintu air ketika diperlukan dan pompa air berfungsi untuk mengalirkan air bersih ke wadah. Motor servo terhubung ke NodeMCU melalui pin power 3V, pin ground, serta kabel signal yang dhiubungkan dengan pin D1. Pompa air yang digunakan memiliki tegangan 5V serta terhubung ke salah satu channel relay dan terhubung ke adapter melalui kabel positif.



Gambar 4. Desain skema perangkat keras

### III. HASIL DAN PEMBAHASAN

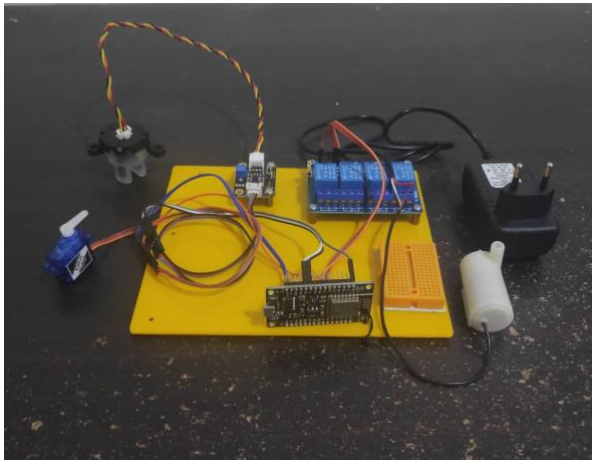
#### A. Implementasi perangkat keras

Pada tahap ini peneliti mengimplementasikan rancangan perangkat keras yang sudah dibuat sebelumnya menjadi sebuah rangkaian perangkat keras elektronika yang dipasang pada wadah berisis air. Bentuk perangkat keras yang diimplementasikan pada penelitian ini dapat dilihat pada gambar 5.

Pada gambar 5 dapat dilihat bahwa NodeMCU yang merupakan perangkat utama dihubungkan dengan perangkat keras lainnya, seperti relay, sensor, motor servo, dan pompa. Perangkat – perangkat ini dihubungkan pada NodeMCU melalui pin – pin yang dimiliki oleh NodeMCU dengan perantara kabel yang sesuai.

#### B. Antarmuka thingspeak

Untuk membuat antarmuka pada *ThingSpeak*, penguji harus terlebih dahulu mendaftar dan kemudian membuat *channel* baru di *ThingSpeak*. Adapun proses pembuatan *channel* baru seperti pada gambar 6.



Gambar 5. Tampak depan alat

Gambar 6. Tampilan saat membuat channel thingspeak

Seperti terlihat pada gambar 6 penguji hanya perlu mengisi nama dari channel dan field yang akan digunakan. Kali ini penguji menggunakan 2 field yaitu field 1 yang diberi nama kekeruhan dan field 2 yang diberi nama pompa. Field 1 akan menampilkan data yang berasal dari sensor sedangkan field 2 akan menampilkan data masukan dari relay yang berfungsi untuk mengatur pompa.

Setelah selesai menciptakan channel baru penguji kemudian menyalin API Key dan ID channel kedalam kode sumber yang akan digunakan. Hal ini guna memberikan alamat yang akurat pada 'thing' untuk mengirimkan data ke channel ThingSpeak yang telah dibuat.

Penguji kemudian menambahkan *widget* berupa indikator dan tampilan numerik untuk menampilkan data dari *field* 1 dan 2. Tampilan dari *channel* yang dibuat dapat dilihat seperti gambar 8 dibawah.

Gambar 8 merupakan tampilan publik dari *channel thingspeak*. Terdapat 2 *widget* pada tampilan publik *channel*, *widget* pertama menampilkan data kekeruhan pada *user* dan *widget* kedua akan menyala ketika pompa dalam keadaan aktif.

#### C. Pengujian sistem

Setelah sistem didesain lalu diimplementasikan, selanjutnya sistem akan diuji apakah sudah bekerja sesuai dengan desain dan bekerja dengan baik. Pada tahap ini sistem akan melewati 3 proses pengujian yaitu, pengujian sensor, pengujian motor servo, pengujian pompa air.

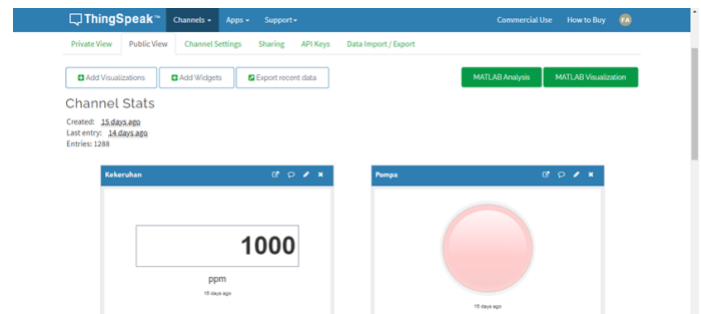
```

AnalogTurbidity_Pompa $
#include <WiFiClient.h>
#include <ThingSpeak.h>
#define relay D1

const char *ssid = "F.G.A";
const char *pass = "A55a_5anger";
const char *server = "api.thingspeak.com";
WiFiClient client;
unsigned long myChannelNumber = 1411586;
const char *myWriteAPIKey = "171ONRRWAME7UKRH";

```

Gambar 7. Kode sumber yang berisi alamat channel thingspeak



Gambar 8. Tampilan public channel thingspeak

1. Pengujian sensor

Pada penelitian ini sensor merupakan perangkat input utama. Untuk memastikan sensor yang digunakan dapat bekerja dengan baik saat sistem dijalankan maka diperlukan pengujian terhadap sensor tersebut. Penguji akan menggunakan kode sumber yang ditampilkan pada gambar 9 untuk melakukan pengujian.

Setelah dilakukan pengujian, sensor dapat bekerja dengan baik dan dapat mengukur kekeruhan pada air dengan baik. Pada gambar 10 terlihat keluaran data sensor pada channel ThingSpeak.

2. Pengujian motor servo

Motor servo pada penelitian ini merupakan perangkat yang digunakan sebagai kontrol pintu air, untuk itu motor servo perlu untuk dievaluasi. Pengujian ini untuk memastikan apakah motor servo dapat bekerja dengan baik dan tidak mengalami masalah. Pengujian akan dilakukan dengan menggunakan kode sumber yang dapat dilihat pada gambar 11.

Setelah kode sumber pada gambar 11 dijalankan, diperoleh hasil bahwa motor servo dapat bekerja dengan baik

3. Pengujian pompa air

Pada penelitian ini pompa air merupakan perangkat output yang digunakan untuk mengalirkan air bersih ke wadah. Pengujian ini untuk memastikan jika pompa bekerja dengan baik dan tidak mengalami masalah. Penguji akan menggunakan kode sumber yang ditampilkan pada gambar 12 untuk melakukan pengujian.

Setelah pengujian menggunakan kode sumber pada gambar 12 dijalankan, diperoleh hasil bahwa pompa air bekerja dengan baik dan tanpa masalah.

Pada gambar 13 terlihat tampilan channel saat menguji pompa. Gambar sebelah kiri adalah keadaan pompa saat mati dan kanan adalah keadaan pompa saat menyala.

4. Pengujian alat keseluruhan

Pada penelitian ini seluruh perangkat output yang telah diuji sebelumnya akan diuji sekali lagi guna memastikan seluruh perangkat output yang ada dapat bekerja secara bersamaan. Penguji akan menggunakan kode sumber yang ditampilkan pada gambar 14 untuk melakukan pengujian.

```

AnalogTurbidity$

void setup() {
  Serial.begin(9600); //Baud rate: 9600 originally
}

void loop(){
  int sensorValue = analogRead(A0); // read the input on analog pin 0:
  float volt = sensorValue * (5.0 / 1023.0);
  float ppm;
  if(volt < 2.5){
    ppm = 1000;
  }else{
    ppm = (-1120.4*(pow((volt),2))+5742.3*volt-4353.9)/3;
  }

  Serial.println(ppm);
  delay(3000);
}
    
```

Gambar 9. Kode sumber untuk menguji sensor

```

Verify

Test_servo

#include <Servo.h>

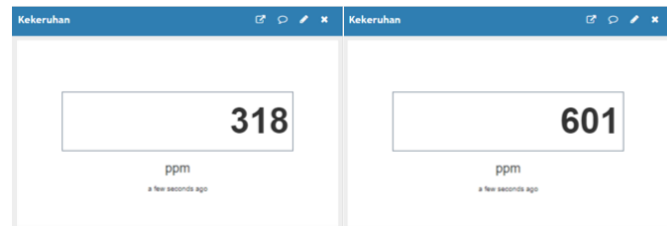
Servo servo;

int pos = 0;

void setup() {
  // put your setup code here, to run once:
  servo.attach (D1);
}

void loop() {
  // put your main code here, to run repeatedly:
  servo.write (pos=90);
  delay (15);
}
    
```

Gambar 11. Kode sumber untuk menguji motor servo



Gambar 10. Tampilan channel saat menguji sensor

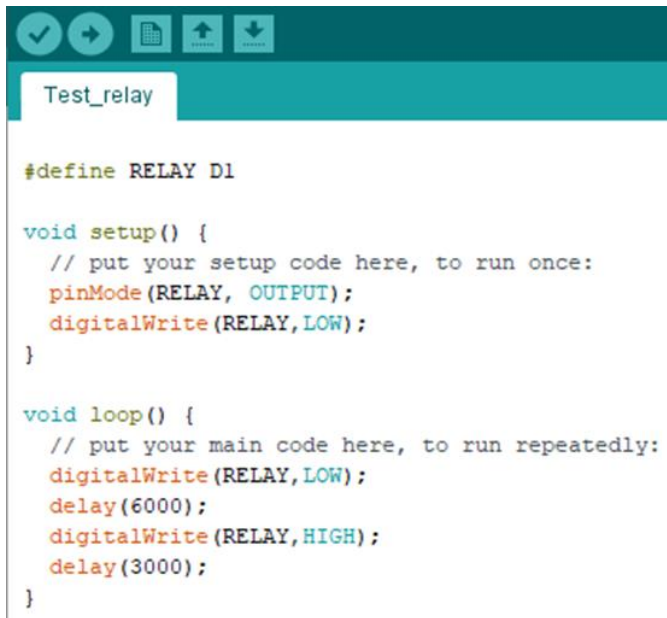
TABEL I  
PENGUJIAN SENSOR

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1.	Turbidity Sensor	Mengukur dan menampilkan hasil bacaan kekeruhan pada monitor	Berhasil

TABEL II  
PENGUJIAN MOTOR SERVO

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1.	Motor Servo SG-90	Motor servo bergerak dan Kembali ke posisi awal	Berhasil





```

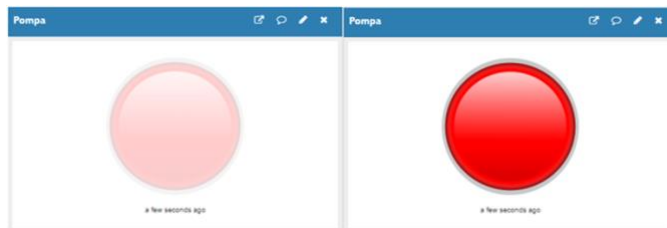
#define RELAY D1

void setup() {
  // put your setup code here, to run once:
  pinMode(RELAY, OUTPUT);
  digitalWrite(RELAY, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(RELAY, LOW);
  delay(6000);
  digitalWrite(RELAY, HIGH);
  delay(3000);
}

```

Gambar 12. Kode sumber untuk menguji pompa air



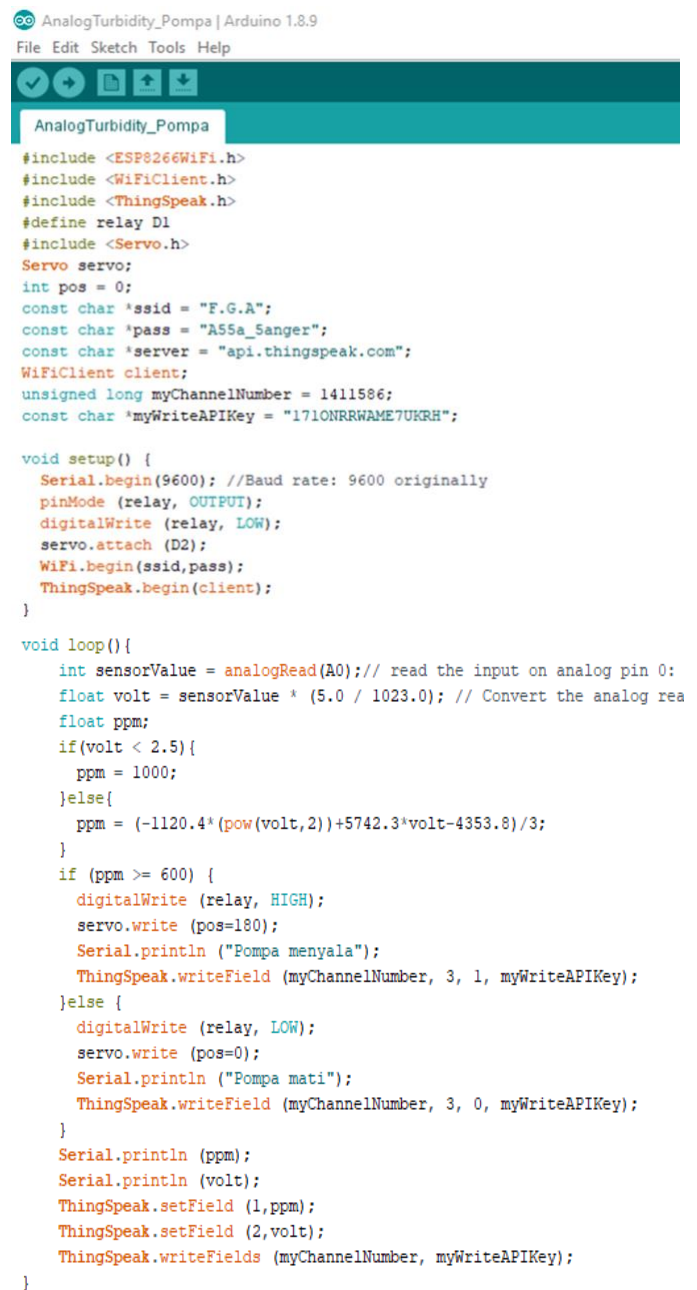
Gambar 13. Tampilan channel saat menguji pompa

TABEL III  
PENGUJIAN POMPA AIR

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1.	Pompa air mini 5V	Pompa dapat merespon pada masukan dari relay dan dapat mengalirkan air	Berhasil

Pada gambar 14 dapat dilihat bahwa pada line 4 sampai 7 merupakan fungsi yang mengatur posisi awal motor servo. Pada line 8 dan 9 merupakan fungsi untuk menghubungkan NodeMCU ke jaringan WiFi dan line 10 sampai 12 merupakan fungsi untuk menghubungkan NodeMCU ke ThingSpeak dan berisi alamat channel ThingSpeak. Pada line ke 17 dan 18 berisi fungsi untuk mengatur kondisi awal relay dan line 19 menunjukkan pin mana yang digunakan untuk menghubungkan NodeMCU dan motor servo. Perintah pada line ke 25 menunjukkan lokasi pin yang menghubungkan sensor dengan NodeMCU. Pada line ke 26 sampai 31 merupakan fungsi untuk menghitung nilai ppm air yang dikonversikan menggunakan formula pada line 31.

Setelah didapati hasil bacaan dari sensor maka berikutnya adalah menentukan aksi yang akan diambil oleh dua perangkat output lainnya berdasarkan hasil bacaan sensor. Fungsi ini



```

AnalogTurbidity_Pompa | Arduino 1.8.9
File Edit Sketch Tools Help

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
#define relay D1
#include <Servo.h>
Servo servo;
int pos = 0;
const char *ssid = "F.G.A";
const char *pass = "A55a_5anger";
const char *server = "api.thingspeak.com";
WiFiClient client;
unsigned long myChannelNumber = 1411586;
const char *myWriteAPIKey = "1710NRRWAME7UKRH";

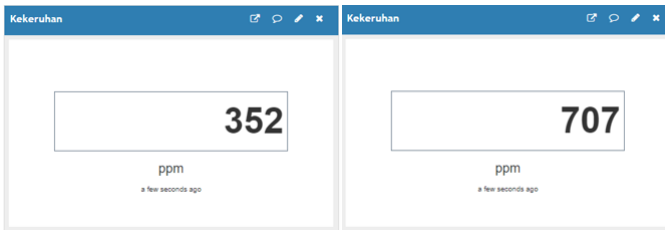
void setup() {
  Serial.begin(9600); //Baud rate: 9600 originally
  pinMode (relay, OUTPUT);
  digitalWrite (relay, LOW);
  servo.attach (D2);
  WiFi.begin(ssid,pass);
  ThingSpeak.begin(client);
}

void loop(){
  int sensorValue = analogRead(A0); // read the input on analog pin 0:
  float volt = sensorValue * (5.0 / 1023.0); // Convert the analog rea
  float ppm;
  if(volt < 2.5){
    ppm = 1000;
  }else{
    ppm = (-1120.4*(pow(volt,2))+5742.3*volt-4353.8)/3;
  }
  if (ppm >= 600) {
    digitalWrite (relay, HIGH);
    servo.write (pos=180);
    Serial.println ("Pompa menyala");
    ThingSpeak.writeField (myChannelNumber, 3, 1, myWriteAPIKey);
  }else {
    digitalWrite (relay, LOW);
    servo.write (pos=0);
    Serial.println ("Pompa mati");
    ThingSpeak.writeField (myChannelNumber, 3, 0, myWriteAPIKey);
  }
  Serial.println (ppm);
  Serial.println (volt);
  ThingSpeak.setField (1,ppm);
  ThingSpeak.setField (2,volt);
  ThingSpeak.writeFields (myChannelNumber, myWriteAPIKey);
}

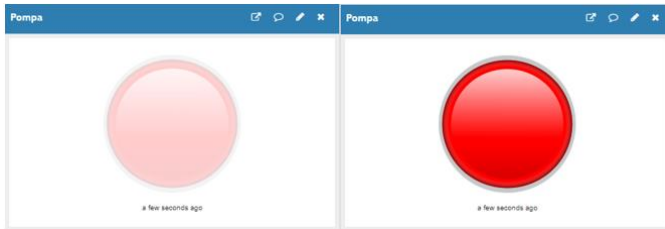
```

Gambar 14. Kode sumber untuk menguji alat secara keseluruhan

dijalankan pada line 33 sampai 42. Fungsi yang terdapat pada line 33 adalah perintah yang akan dijalankan apabila bacaan ppm mencapai nilai lebih dari atau sama dengan 600. Ketika ketentuan dari line 33 terpenuhi maka rantai perintah akan berlanjut ke line 34 dan 35 yang kemudian akan mengubah posisi motor servo dan mengirim sinyal ke relay untuk menyalakan pompa. Pada line 37 merupakan fungsi yang akan mengirimkan data ke field 3 pada channel ThingSpeak. Sebaliknya line 38 sampai 42 berisi perintah apabila nilai bacaan ppm tidak lebih dari 600. Perintah pada line 39 dan 40 membuat motor servo bergerak ke posisi semula dan relay mengirim sinyal untuk mematikan pompa. Nilai bacaan dari sensor akan dikirimkan ke channel ThingSpeak menggunakan fungsi yang terdapat pada line 46 sampai 48.



Gambar 15. Tampilan kekeruhan pada channel saat pengujian alat



Gambar 16. Tampilan pompa pada channel saat pengujian alat

Hasil dari pengujian menggunakan kode sumber pada gambar 14 berjalan dengan baik dan didapati hasil sebagai berikut.

Seperti terlihat dari gambar 15 dan gambar 16, ketiga alat output dapat bekerja secara baik dan pompa serta motor servo dapat berkomunikasi dengan sensor dengan baik.

#### IV. KESIMPULAN DAN SARAN

##### 1. Kesimpulan

Setelah melalui tahapan analisis, desain, implementasi dan pengujian yang telah dilakukan dapat diambil kesimpulan pada penelitian ini sebagai berikut:

- Alat pengukur kekeruhan air akuarium berbasis NodeMCU berhasil dibuat.
- Perawatan akuarium dapat dilakukan secara otomatis dan dapat dipantau secara real-time melalui cloud.
- Berdasarkan hasil pengujian perangkat keras input dan output system yaitu, turbidity sensor, motor servo, dan pompa air dapat bekerja dengan baik.
- Berdasarkan hasil pengujian kekeruhan, alat dapat bekerja dengan baik saat kondisi kekeruhan air mencapai 600 ppm dan berhasil menurunkan kadar ppm air ke kondisi normal.

##### 2. Saran

- Untuk penelitian selanjutnya dapat menggunakan lebih banyak sensor guna mengatur akurasi alat.
- Untuk penelitian selanjutnya dapat menggunakan algoritma yang lebih efisien dalam mendeteksi kekeruhan air.

TABEL IV  
PENGUJIAN ALAT KESELURUHAN

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1	Turbidity Sensor	Mengukur dan menampilkan hasil bacaan kekeruhan pada monitor	Berhasil
2	Motor Servo SG-90	Motor servo bergerak dan Kembali ke posisi awal	Berhasil
3	Pompa air mini 5V	Pompa dapat merespon pada masukkan dari relay dan dapat mengalirkan air	Berhasil

#### V. KUTIPAN

- Indriani, A., Hendra, Witanto, Y., Error of Assembly Microcontroller Arduino Mega and ATmega in the Control of Temperatur for Heating and Cooling Sistem, Applied Mechanics and Materials, ISSN: 1662-7482, Vol. 842, Online: 2016-06-2, Trans Tech Publications, Switzerland.
- Indriani, A., Mesin Pengontrol Temperatur Air Aquarium otomatis Berbasis Mikrokontroller Untuk Budidaya Ikan Kerapu dengan Menggunakan Sumber Energi Terbarukan, Laporan Hibah Bersaing KEMENRISTEKDIKTI, 2015.
- R. Uldin, Pemanfaatan Rangkaian Pengukur Intensitas Cahaya Untuk Rancang Bangun Alat Pengukur Tingkat Kekeruhan Air. Semarang 2011.
- Hidayat, Catur, 2021. Cloud Computing [online]. Available: <https://itkampus.com/cloud-computing>. Diakses 08 Maret 2021.
- Tri Saputro, Tedy, 2017. Mengenal NodeMCU: Pertemuan Pertama [online]. Available: <https://embeddednesia.com/v1/tutorial-nodemcu-pertemuan-pertama>. Diakses 08 Maret 2021. *Examples:*
- Fahad, Engr, 2020. Turbidity Sensor with Arduino for Water Quality Monitoring, Turbidity Meter [online]. Available: <https://www.electronicclinic.com/turbidity-sensor-with-arduino-for-water-quality-monitoring-turbidity-meter/>. Diakses 08 Maret 2021.
- Pelayo, Roland, 2018. Using an Arduino Turbidity Sensor, [online]. Available: <https://www.teachmemicro.com/arduino-turbidity-sensor/>. Diakses 08 Maret 2021.

#### TENTANG PENULIS



**Faentino Gilbert Assa**, Lahir di Manado, pada tanggal 02 November 1996. Anak pertama dari dua bersaudara dari pasangan alm. bapak dr. Felix Nelson Assa M.Kes dan ibu Tineke Like Sanger. Penulis mulai menempuh Pendidikan di Taman Kanak-kanak Paniki Santa Anna (2001-2002). Pendidikan dasar di Sekolah Dasar Santo Petrus Nabire (2002-2008). Pendidikan menengah pertama di SMP Antonius Nabire (2008-2011). Dilanjutkan Pendidikan menengah atas di SMA Adhi Luhur Nabire (2011-2014). Pada tahun 2014 penulis melanjutkan Pendidikan S1 di perguruan tinggi Universitas Sam Ratulangi Manado, Sulawesi Utara. Mengambil Program studi Teknik Informatika Jurusan Elektro Fakultas Teknik. Penulisan Skripsi dengan Judul Pengukur Kekeruhan Air Akuarium Menggunakan Layanan Komputasi Awan IOT sebagai syarat untuk meraih gelar sarjana S1.