

PEMBUATAN SISTEM KONTROL KONDISI *AQUASCAPE* OTOMATIS BERBASIS IOT

EDWIN SYIHAB HARIANTO¹, LIMAN HARTAWAN²

¹Program Studi Teknik Mesin, Institut Teknologi Nasional

²Program Studi Teknik Mesin, Institut Teknologi Nasional
Email : edwinskyihab@gmail.com

Received 16 01 2022 | *Revised* 17 01 2022 | *Accepted* 18 01 2022

ABSTRAK

Mulai banyaknya peminat aquascape namun Ketika peminat aquascape tersebut mempunyai banyak aktivitas padat akan merasa kesulitan dalam merawat aquascape. Pemantauan intensif perlu dilakukan pada aquascape yaitu pada temperature air, lama nyala lampu untuk fotosintesis tanaman, dan pemberian pakan otomatis untuk ikan. Penelitian ini bertujuan untuk menciptakan suatu system control kondisi aquascape otomatis berbasis IoT agar penggemar aquascape yang mempunyai banyak aktivitas padat senantiasa dapat memonitoring dan mengontrol tanaman dan ikan yang ada didalam aquascape dari jarak jauh dan dari manapun tanpa khawatir tanaman dan ikan pada aquascape tersebut mati karena temperature air yang terlalu tinggi, kekurangan pencahayaan, dan kekurangan makanan. Berdasarkan hasil pengujian, system control kondisi aquascape otomatis berbasis IoT mampu menjaga temperature suhu air dengan rentan 20°C s/d 30°C, mampu menyalakan lampu secara otomatis untuk fotosintesis tanaman selama 7 jam per hari, dan mampu memberikan pakan otomatis untuk kebutuhan ikan sebanyak 20 gram setiap 12 jam sekali.

Kata Kunci: *Aquascape, IoT, Temperatur*

ABSTRACT

There are many aquascape enthusiasts, but when the aquascape enthusiast has a lot of solid activities, they will find it difficult to take care of the aquascape. Intensive monitoring needs to be done on the aquascape, namely the water temperature, the length of the lamp for photosynthesis of plants, and automatic feeding for fish. This study aims to create an IoT-based automatic aquascape condition control system so that aquascape enthusiasts who have a lot of solid activity can always monitor and control the plants and fish in the aquascape remotely and from anywhere without worrying about the plants and fish in the aquascape dying due to temperature. too high water, lack of lighting, and lack of food. Based on the test results, the IoT-based automatic aquascape condition control system is able to maintain water temperatures ranging from 20°C to

Edwin, Liman

30°C, is able to turn on the lights automatically for plant photosynthesis for 7 hours per day, and is able to provide automatic feed for fish needs. 20 grams every 12 hours.

Keywords: *Aquascape, IoT, Temperature*

1. PENDAHULUAN

Aquascape adalah seni mengatur dan memelihara tanaman air, batu, batu karang, koral, atau kayu apung, ikan, secara alami dan indah di dalam akuarium sehingga memberikan efek seperti berkebum di bawah air. Tujuan utama *Aquascaping* yaitu untuk menciptakan sebuah pemandangan bawah air yang bagus dengan mempertimbangkan aspek pemeliharaan tanaman air. Bukan hanya ikan yang dijadikan sebagai subjek utama dalam *aquascape*, melainkan tanaman yang menjadi subjek utama. **(Widhianto Harsono:2012).**

Suhu ideal air untuk *aquascape* yaitu berkisar antara 20°C s/d 30°C, dan membutuhkan pencahayaan agar tanaman bisa berfotosintesis dan bisa tumbuh optimal. Pencahayaan yang baik yaitu menggunakan lampu LED (*Light Emitting Diode*) dengan durasi waktu 5 sampai dengan 7 jam per hari. Selain itu kebutuhan pakan ikan juga harus dijaga untuk keberlangsungan hidup ikan pada *aquascape*. **(Fortuna, Tiara Rohma Dewi:2019).**

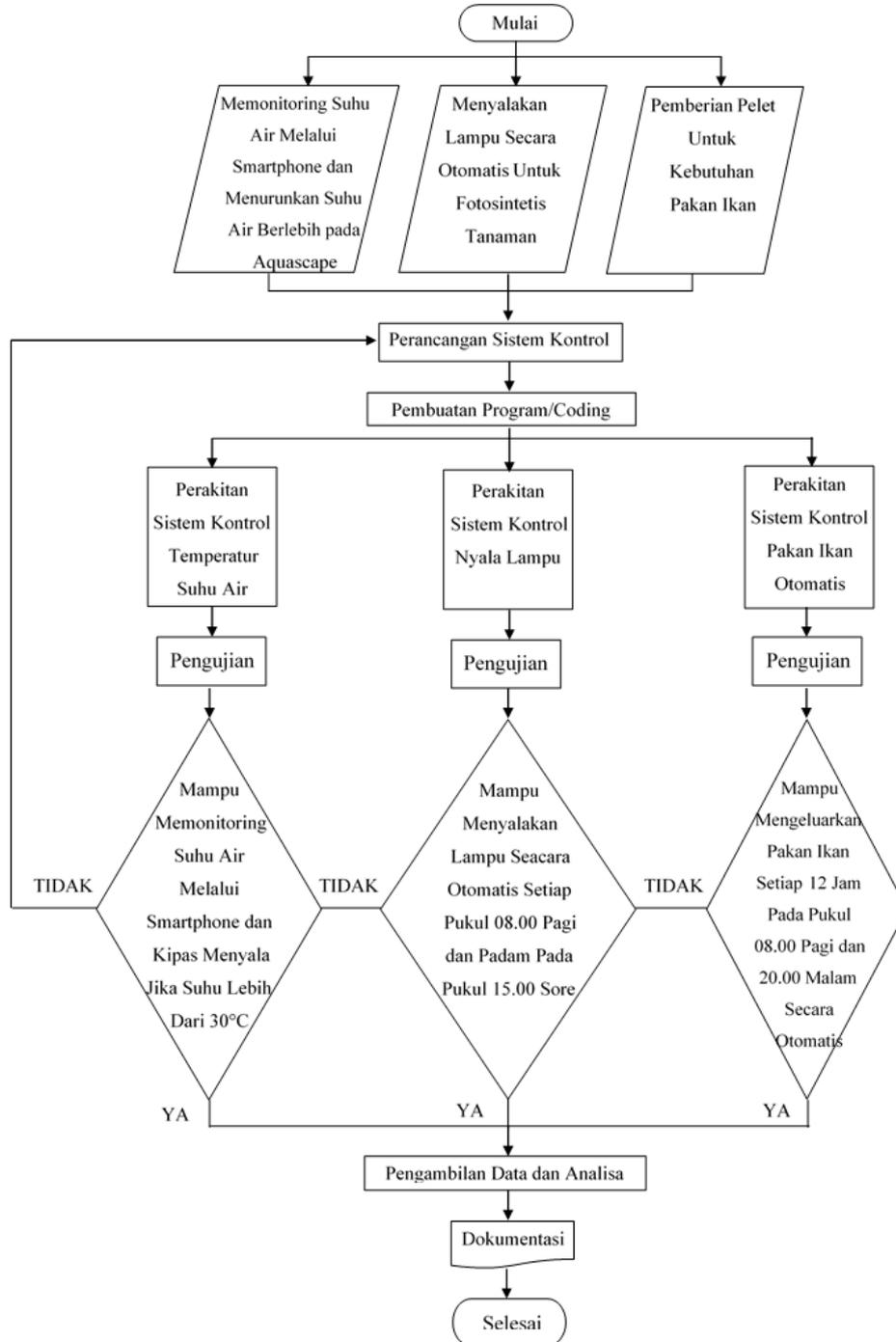
Dampak yang terjadi akibat terlalu tingginya *temperature* suhu pada air yang ditimbulkan oleh pencahayaan dari lampu LED atau perubahan cuaca di lingkungan sekitar bisa menyebabkan tanaman menjadi *melting* atau terlalu panas dan ikan-ikan sulit beradaptasi yang bisa menyebabkan kematian pada tanaman dan ikan. Oleh sebab itu, pemilik *aquascape* harus memperhatikan *temperature* air, lama nyala lampu untuk fotosintesis tanaman, dan pemberian pakan untuk ikan sehingga perlu dilakukan monitoring dan kontrol kondisi *aquascape* secara berkala. Hal tersebut sangat menyibukkan pemilik *aquascape* agar selalu berada di rumah. Ketika pemilik *aquascape* meninggalkan rumah dalam waktu yang lama serta mempunyai banyak aktivitas padat, serta ada perubahan *temperature* suhu air yang signifikan akan berdampak fatal pada tanaman dan ikan.

Untuk mengatasi hal tersebut maka dilakukan perancangan dan pembuatan sistem kontrol kondisi *aquascape* otomatis berbasis IoT yang dapat dipantau dari jarak jauh dan dari manapun. Pembuatan sistem kontrol kondisi *aquascape* otomatis berbasis IoT ini diharapkan dapat memudahkan pekerjaan pemilik atau penggemar *aquascape* dalam memperhatikan *temperature* air, lama nyala lampu untuk fotosintesis tanaman, dan pemberian pakan otomatis untuk ikan karena dapat memantaunya dalam aplikasi *smartphone* sehingga pemilik tidak perlu khawatir jika sedang melakukan pekerjaan lain dan saat ditinggal bepergian.

2. METODOLOGI PENELITIAN

2.1. Diagram Alir Penelitian

Gambar 1 menunjukkan diagram alir penelitian yang terdiri dari proses atau tahapan penelitian sebagai berikut:



Gambar 1. Diagram Alir Penelitian

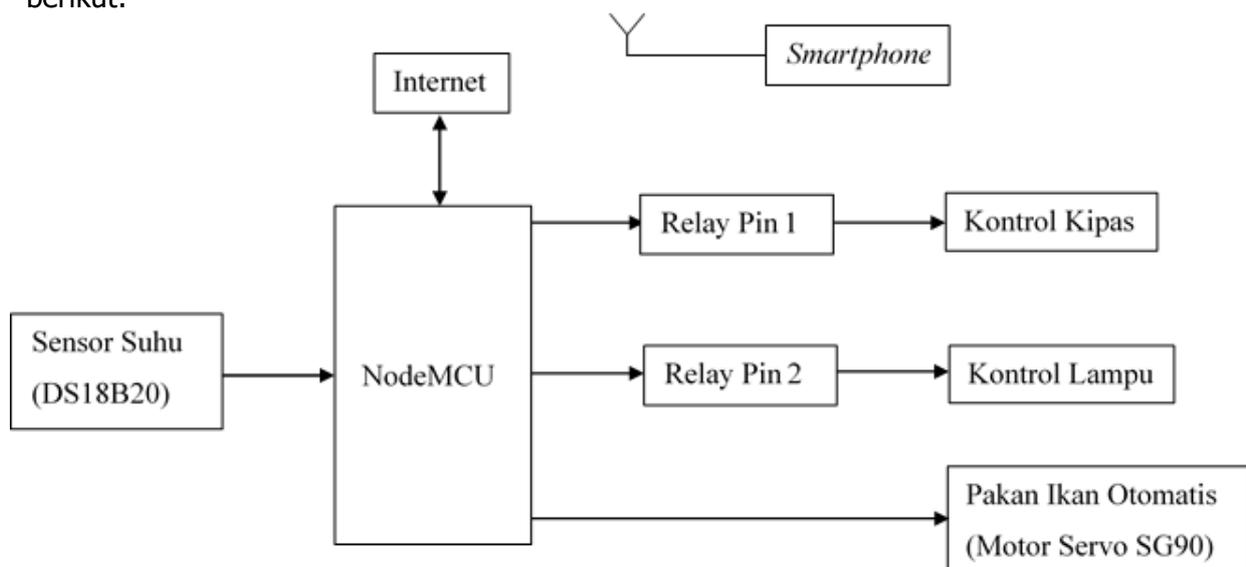
Berdasarkan diagram alir penelitian Gambar 1, proses penelitian memiliki beberapa tahapan pelaksanaan mulai dari inputan atau batasan yang akan dibahas untuk menunjang penelitian hingga pengambilan data dan analisa yang ditunjukkan oleh diagram alir penelitian pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT.

Penelitian dimulai dengan inputan yang yaitu memonitoring suhu air melalui *smartphone* dan menurunkan suhu air berlebih pada *aquascape*, menyalakan lampu secara otomatis untuk fotosintetis tanaman, dan pemberian pelet untuk kebutuhan pakan ikan. Langkah selanjutnya yaitu perancangan *system control* dengan membuat *wiring diagram* menggambarkan alamat / posisi kabel, simbol simbol kelistrikan seperti kontak / saklar, komponen, dll. Setelah perancangan *system control*, langkah selanjutnya yaitu pembuatan program/*coding* menggunakan *software* Arduino IDE dan dilanjutkan dengan perakitan komponen.

Untuk perakitan *system control temperature* suhu air, pengujian dapat dikatakan berhasil jika sensor suhu mampu memonitoring suhu air melalui *smartphone* dan kipas menyala jika suhu lebih Dari 30°C. Untuk perakitan *system control* nyala lampu otomatis, pengujian dapat dikatakan berhasil jika mampu menyalakan lampu secara otomatis setiap pukul 08.00 pagi dan akan padam pada pukul 15.00 sore. Untuk perakitan *system control* pakan ikan otomatis, pengujian dapat dikatakan berhasil jika mampu mengeluarkan pakan ikan setiap 12 jam pada pukul 08.00 pagi dan 20.00 malam secara otomatis dengan berat pakan 20 gram. Setelah proses perakitan dan pengujian selesai, langkah selanjutnya yaitu pengambilan data dan analisa, dan terakhir adalah dokumentasi.

2.2. Blok Diagram Skematik

Gambar 2 menunjukkan blok diagram skematik yang terdiri dari beberapa komponen sebagai berikut:



Gambar 2. Blok Diagram Skematik

Gambar 2 menjelaskan sebuah *system control* kondisi *aquascape* otomatis berbasis IoT menggunakan mikrokontroler NodeMCU yaitu sebagai berikut:

- Bagian Mikrokontroler NodeMCU digunakan untuk menjalankan dan memproses semua perintah *system control*. NodeMCU juga akan menerima sinyal atau perintah dari inputan yaitu dari sensor suhu DS18B20. Selain itu NodeMCU juga bisa menerima sinyal langsung dari *smartphone* untuk menjalankan suatu perintah atau memonitoring *temperature* suhu air dengan catatan NodeMCU perlu terhubung ke internet.
- Sensor *temperature* suhu air DS18B20 sebagai inputan, apabila suhu air melebihi 30°C maka sensor *temperature* suhu air DS18B20 akan mengirimkan sinyal ke NodeMCU, kemudian oleh NodeMCU akan diproses dan mengeluarkan sinyal *output* untuk diteruskan ke relay,

kemudian relay akan menyalakan kipas DC 12V.

- c. Relay sebagai *output*. Relay digunakan untuk memberikan fungsi penundaan waktu, dan untuk menjalankan fungsi logika. Relay akan berjalan ketika mendapatkan sinyal berupa perintah dari NodeMCU kemudian akan diteruskan untuk menyalakan kipas DC 12V dan lampu.
- d. Motor Servo sebagai *output*. Tuas motor servo akan berputar untuk mengeluarkan pakan ikan ketika mendapatkan sinyal berupa perintah dari NodeMCU.
- e. *Smartphone* yang terhubung ke internet dan telah terdapat aplikasi Blynk digunakan untuk memberikan perintah yang bisa berupa menyalakan lampu secara otomatis. Selain itu, *smartphone* yang telah terdapat aplikasi Blynk juga bisa memonitoring *temperature* suhu air pada *aquascape*.

2.3. Perancangan Perangkat Keras Sistem Kontrol Kondisi *Aquascape* Otomatis Berbasis Iot

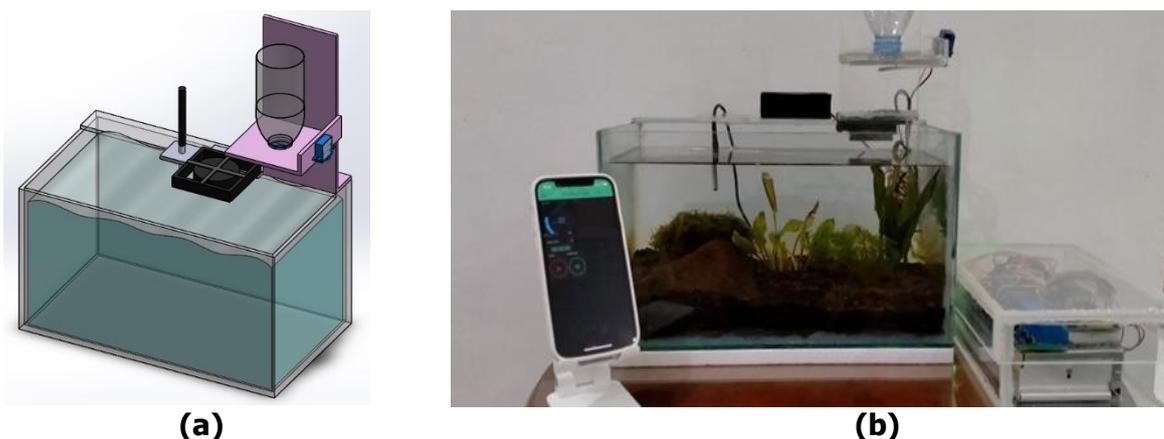
Rangkaian sistem yang dibuat menggunakan beberapa komponen elektronik sebagai pendukung dari seluruh sistem, terdiri dari mikrokontroler NodeMCU, sensor DS18B20, kipas DC, lampu LED, motor servo, relay, dan *power supply*. Untuk perangkat keras yang digunakan memiliki spesifikasi sebagai berikut:

1. Sensor DS18B20 : -10°C sampai +85°C
2. Kipas DC : 12 Volt DC 0,09A
3. Lampu LED : 9 Watt 10000 Kelvin
4. Motor Servo : torque 1.8kg/cm(4.8v) 0° sampai 180°
5. Relay 5 Volt 8 *Channel* : 5 VDC 10A, 250VAC ;15A, 125VAC ; 10A, 250 VAC
6. Step Down 5Volt : Input Voltage DC 3V - 40V; Output Voltage DC 1.5V - 35V
7. *Power Supply* : 12 Volt DC 10A
8. *Breadboard* : 400 hole
9. *Power Supply* : 12 Volt DC 10A

3. HASIL DAN PEMBAHASAN

3.1. Desain Sistem Kontrol Kondisi *Aquascape* otomatis Berbasis IoT

Pembuatan desain 3D *system control* kondisi *aquascape* otomatis berbasis IoT menggunakan *software* solidworks 2015 dan pengaplikasian *system control kondisi aquascape* otomatis berbasis IoT pada *aquascape* dapat dilihat pada gambar 3.



Gambar 3. Desain 3D menggunakan solidworks (a) Pengaplikasian *system control* kondisi *aquascape* otomatis berbasis IoT pada *aquascape* (b)

3.2. Spesifikasi Perancangan Alat Sistem Kontrol Kondisi *Aquascape* otomatis Berbasis IoT

Spesifikasi perancangan *system control* kondisi *aquascape* otomatis berbasis IoT dapat dilihat pada tabel 1.

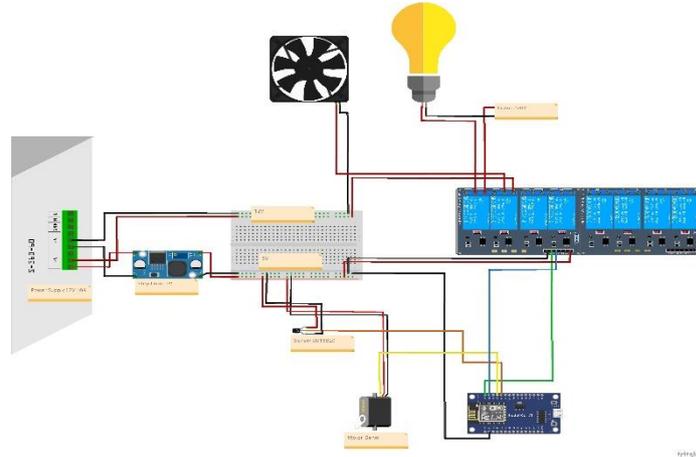
Tabel 1. Spesifikasi Perancangan Alat

NO	Spesifikasi
1	Mampu menjaga <i>temperature</i> suhu air <i>aquascape</i> di angka 20°C hingga 30°C.
2	Mampu menyalakan kipas DC 12V apabila suhu air <i>aquascape</i> lebih dari 30 °C.
3	Mampu menyalakan lampu LED selama 7 jam per hari dimulai pukul 08.00 dan padam pukul 15.00 secara otomatis.
4	Mampu mengeluarkan pakan ikan setiap 12 jam sekali dengan berat pakan 20 gram.

3.3. Perancangan Sistem Kontrol Kondisi *Aquascape* otomatis Berbasis IoT

Pada pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT ini objek yang digunakan untuk penelitian menggunakan aquarium berbahan kaca dengan ukuran panjang 35 cm x lebar 20 cm x tinggi 20 cm dengan adanya tumbuhan dan ikan didalam aquarium tersebut atau biasa disebut *aquascape*. Ada tiga tahapan dalam pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT, diantaranya yaitu pembuatan *system control* menggunakan sensor *temperature* suhu air, pembuatan *system kontrol* menyalakan lampu secara otomatis untuk fotosintetis tanaman, dan pembuatan *system control* pemberian pakan ikan secara otomatis.

Untuk pembuatan *system control* kondisi *aquascape* otomatis berbasis Iot, mikrokontroler yang digunakan untuk menjalankan semua perintah *system control* adalah NodeMCU LUA WiFi V3. Sedangkan aplikasi Blynk pada *smartphone* digunakan untuk memonitoring suhu dan digunakan untuk memberikan perintah untuk menyalakan lampu LED dengan menekan tombol yang ada di aplikasi Blynk pada *smartphone*. *Wiring diagram* pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT dapat dilihat pada gambar 4.



Gambar 4. *Wiring Diagram*

3.4. Pembuatan Program Dengan *Software* Arduino IDE

Pada pembuatan program/ *coding system control temperature* suhu air, perangkat lunak yang digunakan adalah *software* Arduino IDE 1.8.16 sebagai programmer dan bahasa pemrograman yang digunakan adalah bahasa C. Bahasa C menghasilkan objek kode yang sangat kecil dan dieksekusi sangat cepat. *Sketch* program bahasa C yang ditulis menggunakan *software* Arduino IDE kemudian diverifikasi terlebih dahulu oleh program *software* arduino tersebut. Kemudian *sketch* program tersebut di unggah ke dalam mikrokontroler NodeMCU. Gambar 5 menunjukkan program yang dibuat pada *software* Arduino IDE.

```

Sistem_Kontrol_Diseminasi | Arduino 1.8.16
File Edit Sketch Tools Help
Sistem_Kontrol_Diseminasi $
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleTimer.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define BLYNK_PRINT Serial // Comment this out to disable print
#include <Servo.h>

char auth[] = "0YptmQ01Io_JeVzGoXeWu7xqkODZ19Sn";
char ssid[] = "My ASUS";
char pass[] = "edwinsyahab";

SimpleTimer timer;
Servo servo;

#define ONE_WIRE_BUS 2 // DS18B20 on arduino pin2 corresponds to I
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
int temp;

void setup()
{
  pinMode(ph_Pin, INPUT);
  servo.attach(D8); //NodeMCU pin D8
  servo.write(75);
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  DS18B20.begin();
  timer.setInterval(1000L, getSendData);
}

void loop()
{
  timer.run(); // Initiates SimpleTimer
  Blynk.run();
  //Jika suhu lebih dari 30
  if (temp > 30) {
    io.write(pin1, LOW); //menyalakan kipas
    delay(1000);
  }
  else if (temp < 27) {
    io.write(pin1, HIGH); //mematikan kipas
    delay(2000);
  }
  else if (temp < 22) {
    io.write(pin1, HIGH); //mematikan kipas
    delay(1000);
  }
  else {
    delay(2000);
  }
  servo.write(180);
  delay(3000);
  servo.write(175);
}

void getSendData()
{
  DS18B20.requestTemperatures();
  temp = DS18B20.getTempCByIndex(0); // Celcius
  Serial.println(temp);
  Blynk.virtualWrite(V3, temp); //virtual pin V3
}
Done Saving.

```

Gambar 5. Pembuatan program pada *software* Arduino IDE

3.5. Pengujian Sistem Kontrol Temperatur Suhu Air

Pada pengujiannya, peneliti menggunakan *thermometer* air celup biasa sebagai pembanding dari sensor *temperature* suhu DS18B20 yang digunakan. Selain itu, penulis juga menggunakan gelas berisi air hangat (suhu lebih dari 30°C) dan gelas berisi air dengan suhu normal untuk pengujian sensor *temperature* suhu DS18B20 dengan tujuan apakah sensor *temperature* suhu DS18B20 bisa bekerja dengan baik atau tidak. Gambar 6. berikut memperlihatkan pengujian menggunakan air dengan suhu normal (a) pengujian menggunakan air hangat (b)



Gambar 6. Pengujian menggunakan air dengan suhu normal (a) pengujian menggunakan air hangat (b)

Pada pengujiannya, ketika sensor *temperature* suhu DS18B20 dicelupkan ke gelas berisi air dengan suhu normal, sensor DS18B20 bisa membaca dengan akurat karena nilai suhu yang ditampilkan pada aplikasi Blynk sesuai dengan *thermometer* air celup biasa. Lalu, ketika sensor *temperature* suhu DS18B20 dicelupkan ke gelas berisi air hangat (suhu lebih dari 30°C) nilai suhu yang ditampilkan pada aplikasi Blynk mengalami kenaikan sama seperti *thermometer* celup biasa yang juga mengalami kenaikan nilai suhu. Selain itu, pada program/*coding* yang dibuat jika suhu lebih dari 30°C maka kipas DC 12V akan menyala. Ketika sensor DS18B20 dicelupkan gelas berisi air hangat (suhu lebih dari 30°C) kipas DC 12V berhasil menyala dan kipas akan kembali padam jika suhu air sudah kurang dari 27°C. Dari hasil pengujian ini didapatkan bahwa *system control temperature* suhu air yang dibuat sudah berjalan dengan baik. Tabel 2 adalah hasil pengujian *system control temperature* suhu air pada *aquascape*.

Tabel 2. Hasil pengujian *system control temperature* suhu air *aquascape*

NO	Suhu Air Aquascape	Kipas DC 12V
1	22°C	OFF
2	25°C	OFF
3	26°C	OFF
4	30°C	ON
5	34°C	ON
6	39°C	ON

3.6. Pengujian Sistem Kontrol Nyala Lampu Otomatis

Pengujian dilakukan dengan cara menekan tombol "ON" pada aplikasi Blynk dan lampu berhasil menyala. Sedangkan ketika menekan tombol "OFF" pada aplikasi Blynk lampu akan padam. Selain itu penulis juga menguji nyala lampu otomatis menggunakan timer. Lampu berhasil menyala pada pukul 08.00 pagi dan padam pada pukul 15.00 sore. Dari hasil pengujian ini didapatkan bahwa *system control* nyala lampu otomatis yang dibuat sudah berjalan dengan baik.

Gambar 7 Memperlihatkan pengujian ketika tombol ON pada aplikasi Blynk belum ditekan (a) Pengujian dengan cara menekan tombol ON (b).



Gambar 7. Tombol ON pada aplikasi Blynk belum ditekan (a) pengujian dengan menekan tombol ON pada aplikasi Blynk (b)

Tabel 3. adalah hasil pengujian *system control* nyala lampu otomatis. Pengujian dilakukan dengan waktu 24 jam. Dari hasil pengujian, didapatkan lampu LED berhasil menyala selama 7 jam yaitu pada pukul 8 pagi sampai dengan pada pukul 15.00.

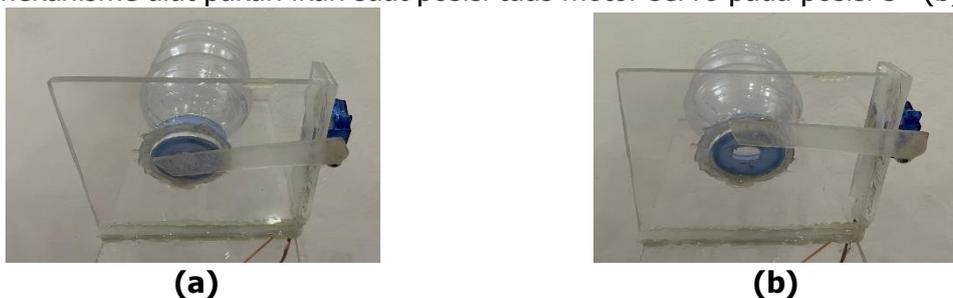
Tabel 3. Pengujian *system control* nyala lampu otomatis

NO	Waktu	Lampu LED
1	06.00	OFF
2	08.00	ON
3	10.00	ON
4	12.00	ON
5	14.00	ON
6	15.00	OFF
7	16.00	OFF
8	18.00	OFF
9	20.00	OFF
10	22.00	OFF
11	24.00	OFF
12	02.00	OFF
13	04.00	OFF

3.7. Pengujian Sistem Kontrol Pemberian Pakan Ikan Otomatis

Pengujian dilakukan dengan cara membuat program *timer* pada *software* Arduino IDE dan alat pemberi pakan otomatis berhasil menyala dan mengeluarkan pakan ikan sebanyak ½ sendok teh (setengah sendok teh) atau setara dengan 20 gram. Selain itu penulis juga melakukan pengujian terhadap program yang telah dibuat sebelumnya yaitu pemberian pakan ikan otomatis setiap 12 jam sekali atau sehari dua kali. Dari hasil pengujian, *system control* pemberian pakan ikan otomatis berhasil mengeluarkan pakan ikan pada pukul 08.00 pagi dan pukul 20.00 malam. Dari hasil pengujian ini didapatkan bahwa *system control* pemberian pakan ikan otomatis yang dibuat sudah berjalan dengan baik.

Gambar 8. memperlihatkan mekanisme pakan ikan saat posisi tuas motor servo pada posisi 0° (a) mekanisme alat pakan ikan saat posisi tuas motor servo pada posisi 5° (b)



Gambar 8. Mekanisme pakan ikan saat posisi tuas motor servo pada posisi 0° (a) mekanisme alat pakan ikan saat posisi tuas motor servo pada posisi 5° (b)

Tabel 4 adalah hasil pengujian alat *system control* pemberian pakan ikan otomatis yang berfungsi dengan baik. Alat pakan ikan mampu mengeluarkan pakan ikan setiap 12 jam sekali dengan rata-rata berat pakan yang dikeluarkan sebanyak 19,69 gram.

Tabel 4. Pengujian *system control* pemberian pakan ikan otomatis

NO	Waktu	Motor Servo	Berat Pakan Yang Dikeluarkan (gram)
1	06.00	ON	18 gr
2	07.00	ON	20 gr
3	08.00	ON	23 gr
4	09.00	ON	16 gr
5	10.00	ON	20 gr
6	11.00	ON	20 gr
7	12.00	ON	16 gr
8	13.00	ON	20 gr
9	14.00	ON	20 gr
10	15.00	ON	22 gr
11	16.00	ON	18 gr
12	17.00	ON	20 gr
13	18.00	ON	23 gr
Rata-rata			19,69 gr

3.8. Analisa

1. Sensor *temperature* suhu air DS18B20 tidak perlu dikalibrasi terlebih dahulu karena pada saat penulis melakukan pengujian, nilai suhu yang dihasilkan oleh sensor suhu DS18B20 sudah akurat dan tepat. Sebagai pembandingan, penulis menggunakan *thermometer* air celup biasa, nilai suhu yang dihasilkan oleh sensor suhu DS18B20 dengan nilai yang dihasilkan oleh *thermometer* air celup biasa yaitu sama.
2. Pada saat pembuatan program untuk pembuatan *system control* sensor *temperature* suhu DS18B20 menggunakan *software* Arduino IDE, perlu di inputkan "delay(1000)" yang berarti mikrokontroler NodeMCU akan selalu memuat ulang *system* atau selalu *refresh* selama satu detik. Jika tidak diinputkan "delay(1000)" NodeMCU akan selalu memuat ulang secara terus menerus sehingga pembacaan nilai sensor suhu terlalu cepat sehingga sulit untuk terbaca.
3. Pada perakitan *system control temperature* suhu air untuk menyalakan kipas dan nyala lampu otomatis, kabel merah pada alat-alat tersebut dimasukan ke pin "NC" atau *Normally Close* relay 8 *channel*. Karena jika kabel merah pada alat-alat tersebut dimasukan ke pin "NO" atau *Normally Open* relay 8 *channel*, alat-alat tersebut akan otomatis menyala tanpa diberikan perintah melalui aplikasi Blynk terlebih dahulu.
4. Pada proses perakitan, jika ingin menghubungkan dua kabel *jumper*, sebaiknya gunakan kabel *jumper* dengan warna yang sama. Hal ini untuk meminimalisir agar tidak terjadi kesalahan dalam menghubungkan kabel *jumper* pada pin yang dituju.
5. Pada proses perakitan, gunakan kabel *jumper* dengan warna yang berbeda pada setiap masing-masing pin pada alat-alat yang digunakan untuk meminimalisir agar tidak terjadi kesalahan dalam menghubungkan *kabel jumper* pada pin satu dan lainnya.

6. Jangan menyimpan alat-alat seperti mikrokontroler NodeMCU, relay 8 *channel*, *power supply* 12V, dan *stepdown* 5V pada tatakan, meja, atau tempat yang beralaskan logam. Karena jika alat-alat tersebut disimpan di tatakan, meja, atau tempat yang beralaskan logam akan membuat IC dari alat-alat tersebut rusak karena logam dapat menghantarkan arus listrik. Lebih baik menggukakan tatakan, atau tempat yang beralaskan kertas, karton, atau kain. Selain itu bisa juga menggunakan meja dari bahan kayu untuk pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT.
7. Perhatikan dan pastikan tegangan *input* pada masing-masing alat yang akan digunakan. Karena jika alat-alat yang akan digunakan menerima tegangan arus yang berlebih atau tidak sesuai dengan spesifikasi, maka dipastikan alat yang akan digunakan tersebut akan rusak. Sedangkan jika nilai tegangan pada alat yang akan digunakan kurang, alat tersebut tidak akan menyala atau tidak bisa berjalan secara maksimal.
8. Wifi atau sinyal yang digunakan untuk koneksi NodeMCU harus stabil, karena jika sinyal atau wifi yang digunakan buruk akan menghambat pemrosesan *system* oleh NodeMCU.

4. KESIMPULAN

1. Berdasarkan hasil pengujian, sensor temperature suhu DS18B20 yang digunakan untuk mengukur suhu air pada *aquascape* dapat bekerja dengan baik. Sehingga jika suhu melebihi angka 30°C kipas DC 12V akan menyala dan kipas akan padam jika suhu sudah dibawah angka 27°C.
2. Berdasarkan hasil pengujian, motor servo yang digunakan yang diterapkan pada system pemberian pakan ikan otomatis dapat berjalan dengan baik. Tuas pada motor servo dapat bergerak atau berputar sebesar 5° searah jarum untuk mengeluarkan pakan ikan sebanyak 20 gram dan akan kembali pada posisi 0° dengan otomatis.
3. Berdasarkan hasil pengujian, relay 8 channel yang digunakan pada pembuatan *system control* kondisi *aquascape* otomatis berbasis IoT dapat bekerja dengan baik. Relay 8 *channel* dapat menjalankan fungsi logika dan penundaan waktu untuk menyalakan kipas DC 12V dan menyalakan lampu untuk pencahayaan pada *aquascape*. Lampu LED berhasil menyala pada pukul 08.00 pagi dan akan padam pada pukul 15.00 sore secara otomatis.
4. Berdasarkan hasil pengujian, aplikasi Blynk yang digunakan pada pembuatan system control kondisi *aquascape* otomatis berbasis IoT dapat bekerja dengan baik. Sehingga bisa menampilkan nilai pengukuran *temperature* suhu air *aquascape*. Selain itu dapat juga membuat program untuk nyala lampu otomatis menggunakan opsi *timer* pada aplikasi Blynk tersebut.
5. Peran pemilik/pengguna *aquascape* telah dapat dikurangi dengan adanya *system control* kondisi *aquascape* otomatis berbasis IoT ini baik dalam proses *control temperature* suhu air, pengaktifan lampu untuk pencahayaan, dan pemberian pakan ikan secara otomatis.

DAFTAR PUSTAKA

Rujukan Jurnal:

- Yohanes Sergio Sili. 2014. Rancang bangun alat pemberian pakan ikan koki otomatis pada aquarium berbasis mikrokontroler AT89S52. Jurnal mahasiswa fakultas Sains dan Teknologi.
- Lestari, Elly. Imtihan, Miftahul. 2020. Perancangan Produk Aquascape Dengan Metode Quality Function Deployment (QFD). Jurnal Terapan Teknik Industri. Vol. 1, No. 1.

Rujukan Prosiding:

- Nuryadi. Haryati. Indrayani, Lilis. 2021. Rancang Bangun Sistem Kendali Keseimbangan Kebutuhan Tanaman Air dalam Aquascape. Proceeding KONIK (Konferensi Nasional Ilmu Komputer) Vol 5.
- Ramadhan, Muhammad Rakha. Handayani, Rini. Sari, Marlindia Ike. 2020. Sistem Monitoring dan Kualitas Air Berbasis Arduino. e-Proceeding of Applied Science. Vol. 6, No. 2.
- Hardyanto, R. Hafid. Ciptadi, Prahenusa Wahyu. 2019. Konsep "AQU PINTAR" Aquarium Pintar 4.0 Berbasis IoT. Prosiding Seminar Nasional Dinamika Informatika Universitas PGRI Yogyakarta. Vol. 3, No. 1.
- Fortuna, Tiara Rohma Dewi. Pangaribuan, Porman. Sumaryo, Sony. 2019. Perancangan Aquarium Pintar Untuk Pemeliharaan Ikan Air Tawar Dengan Algoritma Context Aware Berbasis IoT. eProceedings of Engineering. Vol. 6, No. 2.

Rujukan Online:

- Harsono Widhianto. 2012. Aquascape. www.academia.edu