

**Sistem Pengkondisian Suhu Berbasis *Peltier* Menggunakan Kendali PID Integrasi IoT Optimalisasi
Pertumbuhan *Java Moss Aquascape***

Putra Priangga Dewangkara

Fakultas Teknik, Jurusan Teknik Elektro, Universitas Negeri Surabaya

email : putra.21069@mhs.unesa.ac.id

Bambang Suprianto, Lilik Anifah, Puput Wanarti Rusimanto

Fakultas Teknik, Jurusan Teknik Elektro, Universitas Negeri Surabaya

email : bambangsuprianto@unesa.ac.id, lilikanifah@unesa.ac.id, puputwanarti@unesa.ac.id

Abstrak

Penelitian ini bertujuan untuk menguji efektivitas sistem kontrol suhu dan pencahayaan otomatis terhadap pertumbuhan *Java Moss* (*Taxiphyllum barbieri*) dalam *Aquascape*. *Java Moss* pada *Aquascape* memerlukan suhu rendah dan stabil agar dapat tumbuh optimal. Sistem pendingin suhu dirancang menggunakan kendali PID dengan parameter yang ditentukan melalui metode Ziegler-Nichols dan disempurnakan melalui *fine tuning* ($K_p = 125$, $K_i = 1$, $K_d = 700$), untuk menjaga suhu air pada 25°C. Pengaturan pencahayaan otomatis berbasis NTP mengaktifkan lampu selama 8 jam per hari untuk meniru siklus cahaya alami. Metode yang digunakan adalah rancang bangun dengan pendekatan *Research and Development* (R&D), mencakup perancangan, implementasi, dan pengujian sistem berbasis IoT menggunakan modul *Peltier* dan mikrokontroler. Data pendukung diperoleh dari studi literatur sebagai referensi desain dan analisis. Hasil pengamatan menunjukkan rata-rata pertumbuhan diameter *Java Moss* sebesar 0,68 cm per minggu dengan variasi antar sampel yang kecil, menandakan sistem bekerja secara efektif dalam menciptakan lingkungan tumbuh yang sesuai.

Kata kunci: *Aquascape*, *Java Moss*, Kendali PID, *Peltier*, Sistem Otomasi, *Internet of Things* (IoT), dan Pertumbuhan Tanaman Air.

Abstract

This study aims to evaluate the effectiveness of an automatic temperature and lighting control system on the growth of Java Moss (Taxiphyllum barbieri) in an aquascape environment. Java Moss in aquascape requires a cool and stable temperature for optimal growth. The temperature control system was designed using a PID controller with initial parameters obtained through the Ziegler-Nichols method and refined through fine tuning ($K_p = 125$, $K_i = 1$, $K_d = 700$), maintaining water temperature at a setpoint of 25°C. Lighting is controlled automatically using the Network Time Protocol (NTP), activating the lights for 8 hours per day to simulate a natural light cycle. The research employed an engineering design method with a Research and Development (R&D) approach, involving the design, implementation, and testing of an IoT-based system using a Peltier module and microcontroller. Supporting data were obtained from literature studies as design and analysis references. Observations showed an average Java Moss diameter growth of 0.68 cm per week with minimal variation among samples, indicating the system effectively provides a stable environment for aquatic plant growth.

Keywords: *Aquascape*, *Java Moss*, PID Control, *Peltier*, Automation System, *Internet of Things* (IoT), and Aquatic Plant Growth.

PENDAHULUAN

Dalam beberapa tahun terakhir, minat terhadap pemeliharaan ekosistem akuatik dalam bentuk *Aquascape* semakin berkembang pesat. *Aquascaping* merupakan kegiatan yang menggabungkan seni dan ilmu pengetahuan untuk menciptakan lanskap bawah air yang tidak hanya memiliki nilai estetika, tetapi juga mendukung ekosistem akuatik yang berfungsi secara alami. Hobi ini telah menarik perhatian banyak kalangan karena keindahan yang dihasilkan serta tantangan dalam menciptakan keseimbangan antara elemen-elemen biotik dan abiotik di dalam air.

Aquascape sendiri merupakan seni menata ekosistem bawah air, yang melibatkan penggunaan tanaman, batu,

kayu, dan elemen lain untuk menciptakan lanskap akuatik yang indah dan sehat (Hariyatno et al., 2018). *Aquascape* pada dasarnya bergantung pada beberapa elemen kunci, yaitu tanaman air, ikan, substrat, suhu, dan pencahayaan (Kumar, 2021). Tanaman air berperan penting dalam menjaga kualitas air dan menyediakan oksigen bagi kehidupan akuatik. Tanaman air, yang tidak hanya memberikan keindahan visual tetapi juga mendukung keseimbangan ekosistem akuatik. Salah satu jenis tanaman yang populer dalam *Aquascape* adalah *Java Moss*.

Java Moss (*Taxiphyllum barbieri*), Tanaman ini merupakan jenis lumut yang memiliki penampilan yang unik dengan bentuk daun kecil-kecil dan tekstur yang halus. Daun

Java Moss biasanya berukuran sangat kecil, dengan panjang sekitar 1-2 mm, dan memiliki bentuk oval memanjang yang tumbuh secara acak di sepanjang batang yang tipis (Peter Hiscock, 2003). Tanaman ini sering kali tumbuh secara merambat dan membentuk lapisan tebal yang menyerupai karpet di atas permukaan substrat atau benda keras seperti batu dan kayu di dalam *Aquascape* (Goffinet & Shaw, 2008).

Pertumbuhan optimal *Java Moss* sangat dipengaruhi oleh faktor lingkungan, terutama pencahayaan dan suhu air. *Java Moss* membutuhkan pencahayaan sedang hingga rendah untuk mendukung proses fotosintesisnya, dengan durasi pencahayaan sekitar 8-10 jam per hari. Selain itu, suhu yang ideal untuk pertumbuhan *Java Moss* berada dalam rentang 15°C hingga 35°C (Jose & Joseph, 2022). Suhu yang optimal mendukung pertumbuhan *moss* yang sehat dan berkualitas. Suhu yang tinggi mempercepat metabolisme tanaman, tetapi sering kali menyebabkan tanaman mengalami stres. Dengan metabolisme yang meningkat, tanaman membutuhkan lebih banyak nutrisi dan karbondioksida (Zhang et al., 2020), yang mungkin tidak selalu tersedia di lingkungan akuatik. Akibatnya, tanaman menjadi rentan terhadap kerusakan. Ketika suhu terlalu tinggi, daun dapat mengalami kerusakan, dan fotosintesis menjadi kurang efisien (Hosokawa et al., 2009). Hal ini berpengaruh pada pertumbuhan *moss*.

Kendala dalam mempertahankan kondisi suhu yang optimal ini sering kali menjadi tantangan bagi para penggemar *Aquascape*, terutama dalam hal pengaturan suhu air dan pencahayaan secara manual. Dengan kemajuan teknologi saat ini, *Internet of Things* (IoT) menawarkan solusi melalui sistem otomatisasi yang dapat memonitor dan mengatur suhu serta pencahayaan secara otomatis. Sistem berbasis IoT ini tidak hanya meningkatkan efisiensi dalam perawatan *Aquascape*, tetapi juga membantu menciptakan kondisi yang lebih stabil dan meningkatkan pertumbuhan optimal tanaman seperti *Java Moss*.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem automasi lampu dan pendingin suhu berbasis *peltier* dengan kendali PID yang terintegrasi dengan IoT pada *Aquascape* guna mengoptimalkan pertumbuhan *Java Moss*. Sistem ini diharapkan dapat memberikan solusi efektif bagi para penggemar *Aquascape* dalam menjaga kondisi optimal untuk pertumbuhan tanaman, sehingga meningkatkan estetika dan kesehatan ekosistem bawah air yang diciptakan.

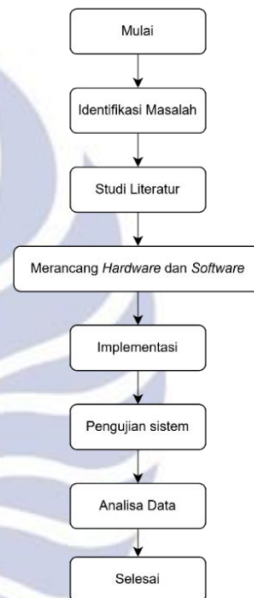
METODE

Penelitian ini menggunakan metode rancang bangun dengan pendekatan *Research and Development* (R&D) untuk mengembangkan dan menguji sistem kendali suhu

terintegrasi IoT. Sistem yang dirancang mengintegrasikan pendingin suhu berbasis *Peltier* yang dikendalikan oleh kendali PID. Penentuan parameter awal PID dilakukan menggunakan metode Ziegler-Nichols, yang kemudian disempurnakan melalui proses *fine tuning* guna memperoleh performa kendali yang optimal.

Tahapan Penelitian

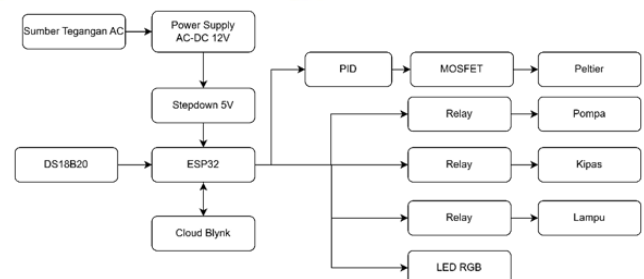
Tahapan penelitian ini dapat dijelaskan pada Gambar 1. Data sekunder yang diperoleh dari studi literatur, seperti jurnal, buku, datasheet, skripsi, serta sumber informasi lain dari internet, akan digunakan sebagai referensi.



Gambar 1. Tahapan penelitian

Perancangan Sistem

Perancangan sistem pada sistem automasi dapat dilihat pada Gambar 2. Diagram blok rancangan sistem yang selanjutnya digunakan sebagai acuan pada implementasi perancangan Hardware. Sedangkan pada pemilihan dan spesifikasi komponen sistem yang akan digunakan dapat dilihat pada Tabel 2. Spesifikasi komponen sistem.



Gambar 2. Tahapan penelitian

Sistem Pengkondisian Suhu Berbasis *Peltier* Menggunakan Kendali PID Integrasi IoT Optimalisasi Pertumbuhan *Java Moss Aquascape*

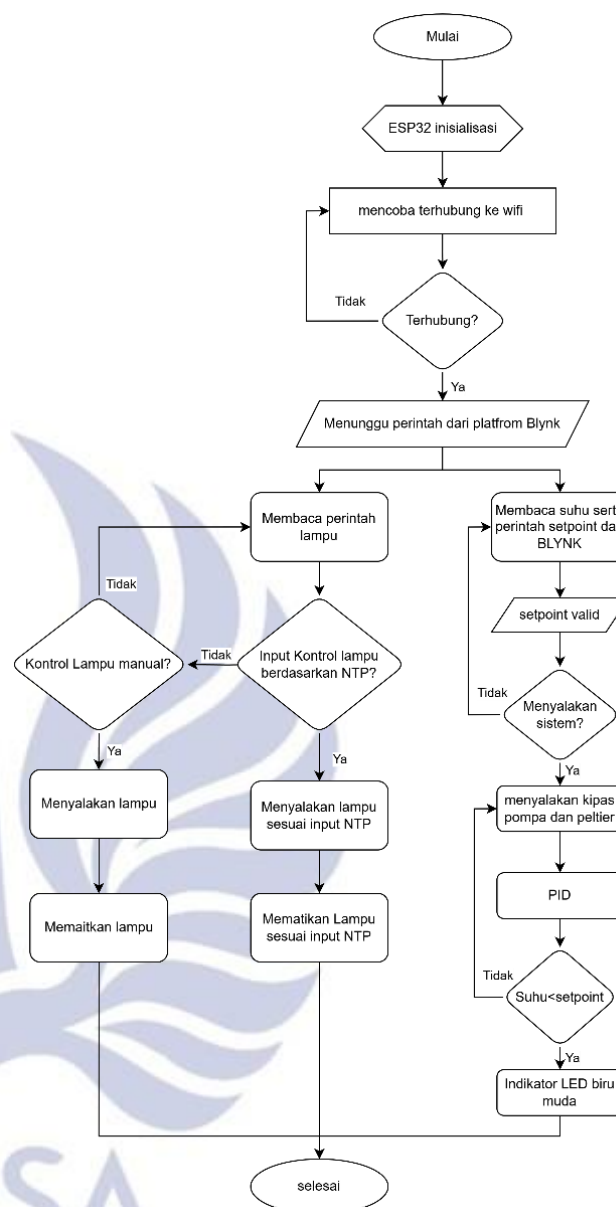
Berdasarkan diagram blok yang ditunjukkan pada Gambar 2 dapat diketahui sistem menggunakan sensor DS18B20 yang berperan sebagai *input*. Data yang diperoleh dari sensor diproses oleh ESP32 yang berperan sebagai mikrokontroler. Hasil pembacaan sensor yang telah diproses oleh ESP32 akan digunakan sebagai acuan untuk menjadi *output* dalam mengaktifkan modul *peltier*, kipas *heatsink* dan pompa pada sistem. Sedangkan untuk lampu menggunakan *Network Time Protocol* (NTP) yang diprogram pada mikrokontroler. Untuk lampu LED sebagai indikator sistem dikontrol sesuai dengan kondisi sistem. Data hasil pemrosesan akan dikirim ke pengguna melalui platform IoT BLYNK, serta ketentuan pengontrolan dalam sistem dapat ditentukan pada platform BLYNK.

Tabel 1. Spesifikasi Komponen

NO	Komponen	Spesifikasi
1	<i>Peltier</i>	Tipe: TEC1-12706. Tegangan kerja 12V, arus 4.5A, daya sekitar 50-60W, dimensi 40x40mm.
2	Mikrokontroler	Tipe: NodeMcu ESP32 Dual-core 32-bit LX6 CPU, frekuensi 240 MHz, Wi-Fi 802.11 b/g/n, Bluetooth 4.2, RAM 520KB, flash hingga 16MB
3	Power supply	Power Supply Switching 12V 10A Sumber tegangan <i>input</i> : 220 Volt AC Tegangan <i>Output</i> : 12V DC Daya maksimal : 10A (120W)
4	Sensor suhu	Tipe: DS18B20. Rentang tegangan : 3.0V to 5.5V Rentang suhu : -55C to +125C
5	Relay	Beban maksimal : AC 250V/10A, DC 30V/10A Tegangan kerja : 5V
6	LED	Sumber tegangan 12V, daya 5W.
7	Heatsink	Dimensi: 15 cm x 10 cm x 4 cm Bahan: Aluminium
8	<i>Waterblock</i>	Dimensi 4 cm x 4 cm x 1,2 cm bahan: aluminium.
9	Pompa DC	Sumber tegangan DC 12V, 0,5-0,7A Aliran 1.2-1.5L/min.
10	MOSFET	Tipe: AOD4184A N-ch, Tegangan 55V, arus 47A, daya 110W
11	StepDown 5V	<i>Input</i> : 12-24V DC (Limit 6~30V DC) <i>Out</i> : 5V DC Max 5 Ampere.
12	Resistor	4.7 k Ω untuk DS18B20 (pull-up resistor).
13	Kipas	Dimensi 80x80x25mm, Tegangan 12V, Arus 0,18A, kecepatan 3000RPM
14	LED RGB	Tipe common anoda 5V

Perancangan Software

Perancangan *software* dapat dilihat pada Gambar 3 perancangan *software*.



Gambar 3. Perancangan software

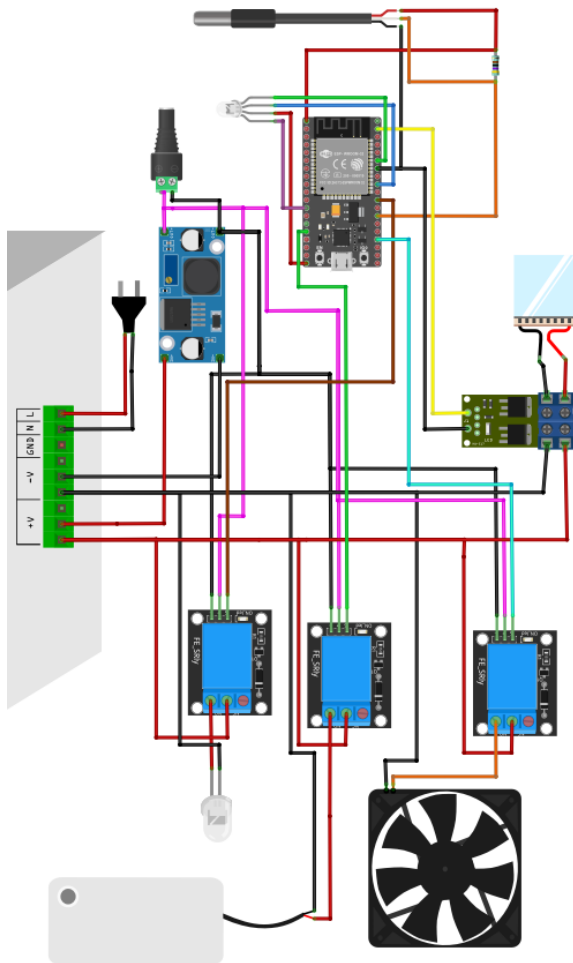
Pemrograman pada sistem automasi lampu dan pendingin diprogram melalui *software* Arduino IDE. Sistem automasi pengaturan lampu dan pendingin suhu pada *Aquascape* berbasis ESP32 bekerja dengan menggabungkan sensor, relay, modul *Peltier*, dan platform Blynk untuk monitoring jarak jauh. Proses dimulai dengan ESP32 yang memproses data dari sensor suhu DS18B20 serta menerima perintah dari platform Blynk. Sistem kemudian mengecek waktu melalui *Network Time Protocol* (NTP) untuk menyalakan lampu secara otomatis sesuai jadwal dan juga mengontrol lampu secara manual ataupun otomatis selama 8 jam. Jika durasi waktu terpenuhi, relay akan diaktifkan untuk

menyalakan lampu, yang akan menyala selama periode tersebut. Setelah durasi berakhir, lampu akan dimatikan.

Selain itu, sensor DS18B20 terus memantau suhu air. Jika suhu air melebihi *setpoint* yang ditetapkan, modul *Peltier* akan diaktifkan untuk menurunkan suhu secara perlahan dengan kendali pwm yang dikontrol oleh PID. Ketika *Peltier* menyala, pompa juga diaktifkan untuk memompa air ke dalam *waterblock* untuk proses pendinginan oleh *peltier*, dan kipas dinyalakan untuk menurunkan suhu pada heatsink sisi panas *peltier*. Sistem akan terus memonitor suhu hingga kembali sesuai dengan *setpoint*.

Implementasi

Setelah perancangan *hardware* dan *software* selesai, tahap ini melibatkan perakitan seluruh komponen menjadi sebuah sistem utuh.



Gambar 4. Skema Perancangan Sistem

HASIL DAN PEMBAHASAN

Hasil dan pembahasan dalam penelitian ini mencakup pengujian setiap komponen sistem, pengujian protokol NTP,

proses tuning kendali PID, serta pengujian performa sistem pendingin suhu dan pencahayaan otomatis pada *Aquascape* yang ditujukan untuk mendukung pertumbuhan *Java Moss*.

Pengujian Sistem

Pengujian dalam penelitian ini dilakukan secara bertahap, dimulai dari pengujian komponen utama sistem seperti sensor suhu dan modul *Peltier* untuk memastikan fungsionalitas masing-masing. Selanjutnya dilakukan pengujian integrasi protokol *Network Time Protocol* (NTP) guna mengatur siklus pencahayaan otomatis sesuai waktu nyata. Proses tuning parameter kendali PID diawali dengan metode Ziegler-Nichols dan disempurnakan melalui *fine tuning* untuk mendapatkan respons sistem yang optimal. Setelah itu, sistem secara keseluruhan diuji pada lingkungan *Aquascape* guna memantau performa pengendalian suhu serta kestabilannya. Terakhir, dilakukan pengamatan pertumbuhan *Java Moss* sebagai indikator biologis untuk menilai efektivitas sistem dalam menciptakan kondisi lingkungan yang mendukung pertumbuhan tanaman secara optimal.

a) Pengujian komponen

Pengujian Sensor DS18B20

Penggunaan sensor DS18B20 pada penelitian ini memiliki peran penting guna membaca suhu air pada *Aquascape* saat proses pendinginan. Untuk mengetahui keakuratan sensor DS18B20 dilakukan perbandingan dengan sensor termometer digital.

Tabel 2. Pengujian Sensor DS18B20

NO	Sensor	Thermometer	Selisih
1	26.3	26.5	0.2
2	27.1	27.4	0.3
3	28.6	28.8	0.2
4	29.2	29.3	0.1
5	30.4	30.6	0.2
6	31.0	31.2	0.2
7	32.3	32.6	0.3
8	33.1	33.3	0.2
10	35.0	35.2	0.2
Jumlah			1,9

Berdasarkan pengujian dapat dihitung nilai error sebagai berikut:

$$\frac{1,9}{10} \times 100\% = 0,19\% \dots \dots \dots (1)$$

Pengujian *Peltier*

Peltier pada penelitian ini digunakan sebagai elemen pendingin suhu air pada *Aquascape*. Pada pengujian ini *Peltier* diaktifkan oleh modul MOSFET yang akan diaktifkan selama beberapa menit setelah itu dimatikan dalam beberapa menit.

Tabel 3. Pengujian *Peltier*

Percobaan Pertama			Percobaan kedua		
menit	<i>Peltier</i>	PWM	menit	<i>Peltier</i>	PWM
1	Aktif	255	1	Mati	0
2	Aktif	255	2	Mati	0
3	Mati	0	3	Aktif	255
4	Mati	0	4	Aktif	255
5	Aktif	255	5	Mati	0

Pengujian kipas

Dalam penelitian ini, kipas digunakan sebagai alat bantu digunakan sebagai kompensasi panas untuk menghilangkan panas dari sisi *hot plate Peltier*. Pada pengujian, kipas diaktifkan secara berkala selama beberapa detik, kemudian dimatikan dalam jeda waktu tertentu.

Tabel 4. Pengujian Kipas

Detik	Kipas	Detik	Kipas
1	Aktif	6	Mati
2	Aktif	7	Mati
3	Aktif	8	Mati
4	Aktif	9	Mati
5	Aktif	10	Mati

Pengujian pompa

Pompa digunakan untuk mengalirkan air menuju sisi dingin dari heatsink *Peltier*. Aliran ini memungkinkan terjadinya perpindahan panas dari air ke permukaan pendingin, sehingga suhu air dapat diturunkan secara efektif. Pada tahap pengujian, pompa diaktifkan dan dimatikan secara berkala dalam interval waktu tertentu.

Tabel 5. Pengujian Pompa

Menit	Pompa	Menit	Pompa
1	Aktif	6	Aktif
2	Aktif	7	Mati
3	Mati	8	Mati
4	Mati	9	Aktif
5	Aktif	10	Aktif

Pengujian Lampu

Lampu digunakan sebagai sumber cahaya utama untuk mendukung pertumbuhan tanaman air melalui proses fotosintesis. Pada tahap pengujian, lampu dinyalakan dan dimatikan dalam durasi waktu tertentu secara bergantian

untuk meniru pengendalian intensitas pencahayaan berdasarkan interval waktu.

Tabel 6. Pengujian Lampu

Menit	Lampu	Menit	Lampu
1	Aktif	6	Mati
2	Mati	7	Aktif
3	Aktif	8	Mati
4	Mati	9	Aktif
5	Aktif	10	Mati

b) Pengujian Protokol NTP

Pengujian dilakukan untuk memastikan sinkronisasi waktu melalui protokol *Network Time Protocol* (NTP) dalam mengatur siklus nyala dan mati lampu secara otomatis. Sistem mengambil waktu dari server NTP dan menyalakan lampu sesuai jadwal yang telah ditentukan. Dengan pengujian ini, dapat dievaluasi apakah penyalakan lampu berjalan sesuai waktu *real-time*.

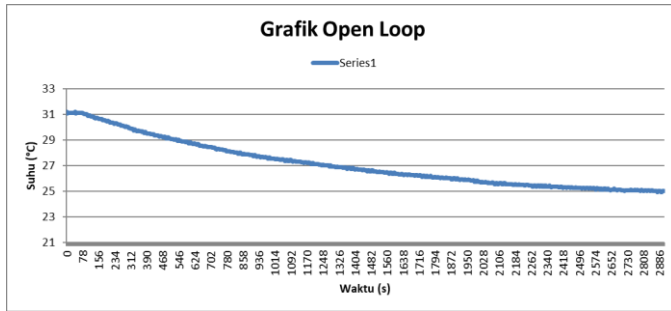
Tabel 7. Pengujian Protokol NTP

NO	Waktu NTP	Status yang diharapkan	Status aktual	Keterangan
1	12.10 PM	Menyala	Menyala	Sesuai
2	12.15 PM	Mati	Mati	Sesuai
3	12.20 PM	Menyala	Menyala	Sesuai
4	12.25 PM	Mati	Mati	Sesuai
5	12.30 PM	Menyala	Menyala	Sesuai
6	12.35 PM	Mati	Mati	Sesuai
7	12.40 PM	Menyala	Menyala	Sesuai
8	12.45 PM	Mati	Mati	Sesuai
9	12.50 PM	Menyala	Menyala	Sesuai
10	12.55 PM	Mati	Mati	Sesuai

c) Pengujian Parameter PID

Pengujian Sistem Secara *Open Loop*

Pengujian *open loop* dilakukan untuk mengetahui respons alami sistem tanpa adanya pengendali. Tahap ini bertujuan untuk memahami karakteristik dinamis sistem, seperti waktu tunda dan konstanta waktu, yang menjadi dasar dalam proses tuning PID. Pengujian dilakukan dengan cara memberikan respon step pada sistem hingga mencapai keadaan stabil. Grafik *open loop* dapat dilihat pada Gambar 5.

Gambar 5. Grafik *Open Loop*

Berdasarkan gambar sistem dapat mencapai suhu 25 derajat dengan suhu awal 31,19 setelah mendapatkan dua nilai suhu tersebut dapat menentukan model matematika dari sistem.

$$\Delta T = 25 - 31.19 = -6.19 \dots\dots\dots (2)$$

$$K = \frac{\Delta T}{\Delta Q} = \frac{25-31.19}{255} = \frac{-6.19}{255} = -0,0243 \dots\dots\dots (3)$$

- Waktu suhu mencapai 63% dari perubahan

$$C_{(t)} = 63\% \times \Delta t = -3.90^\circ C \dots\dots\dots (4)$$

- Tambahkan pada suhu awal

$$T_T = T_{awal} + C_{(t)} = 31.19 - 3.90 = 27.29^\circ C \dots\dots\dots (5)$$

- Cari waktu saat suhu 27.29°C

$$\text{Waktu } T = 1137 \text{ detik} \dots\dots\dots (6)$$

- Cari fungsi transfer

$$G_{(s)} = \frac{C_{(t)}}{R_{(s)}} = \frac{k}{T_s + 1} = \frac{-0.0243}{1137s + 1} \dots\dots\dots (7)$$

Keterangan:

K = Gain

ΔT = Perubahan suhu akhir – suhu awal

ΔQ = Perubahan *input* PWM

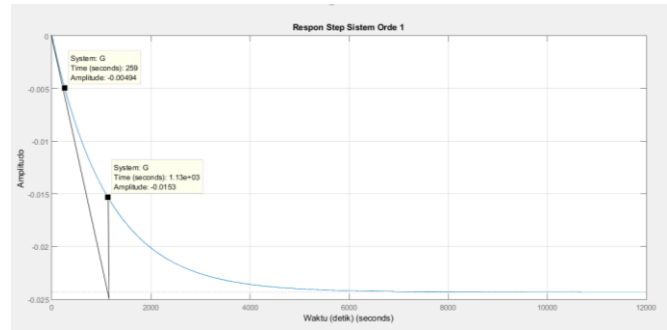
T = Waktu suhu saat 63% atau Waktu Konstan

$C_{(t)}$ = Suhu saat 63% keadaan tunak

Setelah mendapat fungsi transfer sistem dilanjutkan untuk melakukan tuning pada MATLAB untuk mencari nilai K_p , K_i , dan K_d .

Pengujian *Tuning* pada MATLAB

Pengujian ini dilakukan menggunakan perangkat lunak MATLAB dengan bantuan Simulink dan alat bantu PID Tuner. Melalui simulasi, diperoleh nilai awal parameter K_p , K_i , dan K_d berdasarkan model matematis sistem yang telah diperoleh dari pengujian *open loop*.



Gambar 6. Grafik respon pada MATLAB

Berdasarkan Gambar 6 nilai $T=1137$ detik dan nilai $L=259$ detik setelah mendapatkan nilai tersebut dilakukan tuning nilai pid mengacu pada Tabel 2.2 sehingga didapat:

$$K_p = 1.2 \times \frac{T}{HL} = 1.2 \times \frac{1137}{259} = 5.27 \dots\dots\dots (8)$$

$$T_i = 2 \times L = 2 \times 259 = 518 \dots\dots\dots (9)$$

$$T_d = 0.5 \times L = 0.5 \times 259 = 129.5 \dots\dots\dots (10)$$

Setelah mendapatkan nilai T_i dan T_d selanjutnya mencari nilai parameter K_i dan K_d sebagai berikut:

$$K_i = \frac{K_p}{T_i} = \frac{5.2}{1748} = 0.0102 \dots\dots\dots (11)$$

$$K_d = K_p \times T_d = 5.2 \times 437 = 682,77 \dots\dots\dots (12)$$

Keterangan

K_p = Konstanta Proportional

T_i = Waktu Integral

T_d = Waktu Derivative

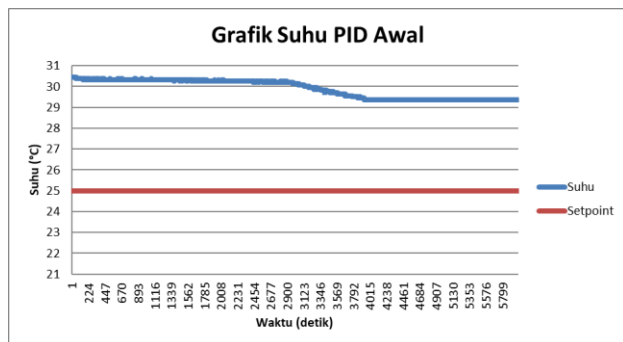
K_i = Konstanta Intergral

K_d = Waktu Derivative

Berdasarkan persamaan diatas didapatkan nilai konstanta $K_p=5.27$, $K_i=0.0102$, $K_d=682,77$. Nilai tersebut akan dilakukan pengujian pada program ESP32 sebagai parameter kontrol PID.

Pengujian *Tuning* Awal PID

Setelah mendapatkan parameter awal dari simulasi, tuning awal dilakukan secara langsung pada sistem nyata untuk mengamati respons awal sistem terhadap hasil parameter awal kendali PID. Tujuannya adalah menyesuaikan hasil simulasi dengan kondisi *real* sistem. Hasil pengujian dapat dilihat pada Gambar 7



Gambar 7. Grafik Parameter PID Awal

Setelah mendapatkan respon dari sistem, maka langkah selanjutnya adalah menganalisa respon sistem tersebut berdasarkan beberapa indikator:

- *Overshoot*

$$M_p = \frac{Y_{ss} - Y_{max}}{Y_{ss}} \times 100\% \quad (13)$$

- *Error steady state*

$$E_{ss} = \text{Setpoint} - \text{nilai suhu akhir} \quad (14)$$

- *Undershoot*

$$U_p = \frac{Y_{min} - Y_{ss}}{Y_{ss}} \times 100\% \quad (15)$$

- *Rise time*

1. Menghitung 10% dan 90% kenaikan nilai

$$T_{10\%} = T_0 + 0,1 \times (T_{ss} - T_0) \quad (16)$$

$$T_{90\%} = T_0 + 0,9 \times (T_{ss} - T_0) \quad (17)$$

2. Identifikasi waktu saat respon mencapai

$$10\% (t_1) \text{ dan } 90\% (t_2). \quad (18)$$

3. Mencari rise time

$$T_r = t_1 - t_2 \quad (19)$$

- *Settling time*

Settling time adalah waktu dimana sistem mencapai kestabilan pada *setpoint*.

Keterangan

MP = Presentase *Overshoot*

y_{max} = Nilai Puncak Awal

y_{ss} = Nilai *Setpoint*

y_{min} = Nilai Minimum Sistem Sebelum Stabil

E_{ss} = *Error steady state*

U_p = Presentase *Undershoot*

T_r = *Rise Time*

TSS = Nilai Suhu Referensi

T₀ = Nilai Suhu Awal

Berdasarkan hasil pengujian tuning awal pada Gambar 7 dengan metode PID Ziegler-Nichols, dapat diambil

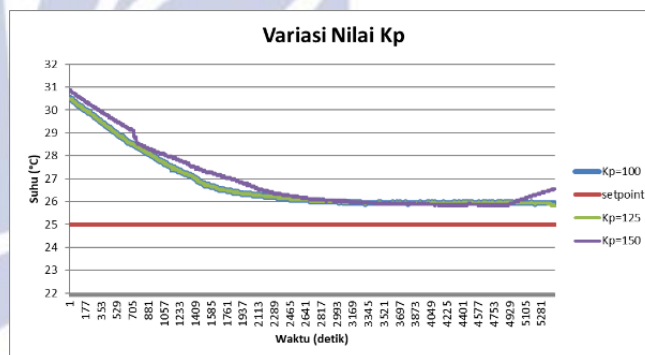
kesimpulan sistem dapat mencapai kestabilan dalam mengendalikan suhu tanpa adanya osilasi. Namun, sistem masih belum optimal karena terdapat *error steady state* sebesar 4,37 °C yang menjadi faktor suhu tidak mencapai *setpoint* yang diinginkan. Oleh karena itu, dilakukan penyesuaian parameter PID lebih lanjut untuk mengurangi *error steady state* dan meningkatkan akurasi sistem dalam mencapai suhu target.

Pengujian *Fine tuning*

Fine tuning dilakukan untuk menyempurnakan respons sistem. Nilai K_p, K_i, dan K_d disesuaikan secara bertahap hingga mendapat respon sistem yang dapat menuju *setpoint* dengan baik tanpa adanya osilasi.

Variasi nilai K_p

Pada pengujian ini, nilai parameter propotional akan divariasikan untuk mendapat respon sistem yang diinginkan. Nilai parameter K_p yang akan dilakukan pada pengujian berjumlah 3 yaitu 100, 125, dan 150. Sedangkan untuk nilai K_i, dan K_d masih menggunakan hasil nilai *tuning* awal yaitu K_i=0.0102 dan K_d=682,77.

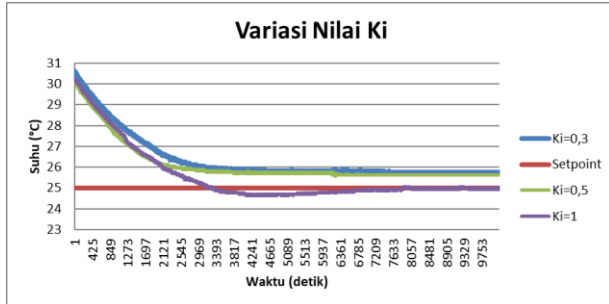


Gambar 8. Grafik Parameter Variasi Nilai K_p

Berdasarkan pengujian parameter K_p Pengujian dengan K_p=100 menghasilkan *overshoot* 0%, *error steady state* 0,94°C, dan *rise time* 444 detik. Pada K_p=125, *overshoot* tetap 0%, *error steady state* menurun menjadi 0,81°C, dan *rise time* 440 detik. Sedangkan pada K_p=150, *overshoot* masih 0%, namun *error steady state* meningkat menjadi 1,56°C dengan *rise time* 464 detik. Sehingga dapat disimpulkan bahwa sistem dengan nilai K_p=125 memiliki respon sistem paling baik dengan nilai *error steady state* 0,81°C lebih rendah dari nilai *propotional* 100 dan 150. Nilai K_p=125 dapat dijadikan nilai referensi pengujian selanjutnya karena respon sistem memiliki *rise time* 440s yang dimana *rise time* lebih cepat dibandingkan dengan nilai *proportional* 100 dan 150.

Variasi nilai K_i

Pada pengujian Ki dilakukan pengujian dengan 3 nilai parameter Ki yaitu 0.3, 0.5, dan 1. Sedangkan untuk nilai Kp menggunakan nilai $K_p=125$ berdasarkan pengujian sebelumnya dan nilai Kd tetap menggunakan nilai *tuning* awal yaitu, $K_d=682,77$.

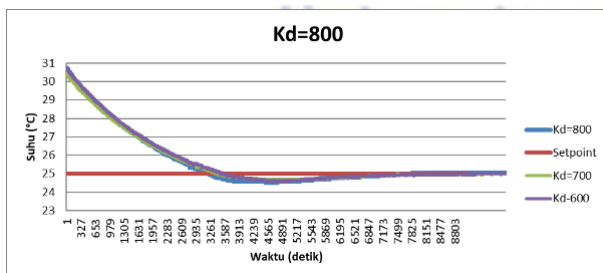


Gambar 9. Grafik Parameter Variasi Nilai Ki

Berdasarkan hasil pengujian dengan $K_i=0.3$, sistem menghasilkan *overshoot* 0%, *error steady state* 0,75°C, dan *rise time* 449 detik. Dengan $K_i=0.5$, diperoleh *overshoot* 0%, *error steady state* 0,25°C, dan *rise time* 455 detik. Sedangkan pada $K_i=1$, sistem mengalami *overshoot* 1,48%, *error steady state* 0,6°C, dan *rise time* 464 detik. Mengacu pada hasil pengujian dapat diambil kesimpulan nilai $K_i=1$ menghasilkan nilai *error steady state* 0,6 °C yang berarti sistem dapat stabil mendekati *setpoint* meski terdapat *overshoot* 1,48% dan *undershoot* 0,6%. Nilai $K_i=1$ dapat dijadikan referensi pengujian selanjutnya sebab dibandingkan nilai parameter integral 0.3 dan 0.5 yang memiliki nilai *error steady state* lebih tinggi dan tidak mencapai *setpoint*.

Variasi Nilai Kd

Pada pengujian Kd digunakan 3 nilai parameter derivative yaitu 800, 700, 600. Untuk nilai Kp dan Ki menggunakan nilai $K_p=125$ dan $K_i=1$ berdasarkan hasil pengujian sebelumnya.



Gambar 10. Grafik Parameter Variasi Nilai Kd

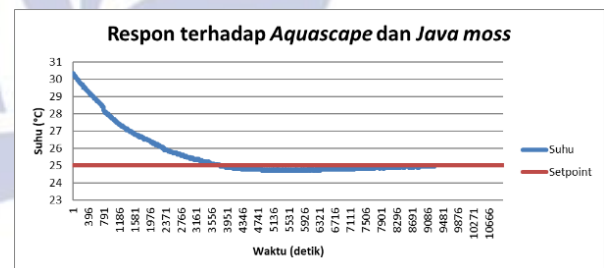
Berdasarkan pengujian dengan $K_d=800$, sistem menghasilkan *overshoot* 2%, *error steady state* 0,6°C, *rise time* 455 detik, dan *settling time* 7877 detik. Dengan $K_d=700$, *overshoot* turun menjadi 1,48%, *error steady state* 0°C, *rise*

time 429 detik, dan *settling time* 7923 detik. Sedangkan pada $K_d=600$, *overshoot* sedikit meningkat menjadi 1,76%, *error steady state* tetap 0°C, *rise time* 455 detik, dan *settling time* 8140 detik. Dari hasil pengujian dapat disimpulkan parameter $K_d=700$ menghasilkan sistem yang stabil dengan *steady state* 0°C yang berarti sistem berjalan stabil dengan *settling time* 7923s lebih lambat daripada nilai parameter $K_d=800$ tetapi, parameter $K_d=700$ memiliki *rise time* yang lebih cepat dibandingkan dengan nilai parameter Kd 800 dan 600. Nilai parameter $K_d=700$ dapat dijadikan sebagai acuan untuk pengujian lanjut meskipun memiliki nilai *overshoot* 1,48% yang masih dalam batas wajar dengan *undershoot* 0%.

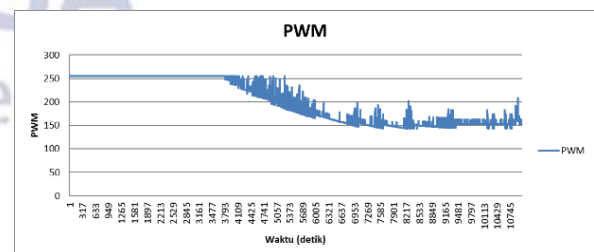
Dengan demikian pada pengujian *fine tuning* didapatkan bahwa parameter terbaik adalah $K_p=125$, $K_i=1$, $K_d=700$. Nilai parameter tersebut dijadikan acuan untuk pengujian berikutnya dengan memasukkan nilai parameter melalui ESP32 untuk mengontrol optimalisasi suhu pada sistem *Aquascape*.

Pengujian Sistem Pada Aquascape

Pengujian sistem pada *Aquascape* dilakukan pada kondisi akuarium yang sudah terisi dengan pasir, batu dan juga *Java Moss*. Pengujian ini bertujuan untuk mengetahui respon sistem terhadap *Aquascape* dan *Java Moss*.



Gambar 11 Grafik Suhu Respon Sistem Terhadap *Aquascape* dan *Java Moss*



Gambar 12 Grafik Perubahan Nilai PWM Selama Proses Kendali

Hasil pengamatan terhadap grafik *output* sinyal PWM dan grafik respon suhu terhadap waktu menunjukkan kinerja sistem kendali PID dalam proses pendinginan air *Aquascape*. Pada grafik PWM, nilai sinyal langsung meningkat ke level maksimum (255) pada awal proses, yang mencerminkan

Sistem Pengkondisian Suhu Berbasis *Peltier* Menggunakan Kendali PID Integrasi IoT Optimalisasi Pertumbuhan *Java Moss* *Aquascape*

respon cepat kontrol PID terhadap selisih suhu yang besar antara suhu awal dan *setpoint*. Hal ini menunjukkan bahwa parameter proporsional (*Kp*) bekerja secara agresif dalam

mengaktifkan modul *Peltier* guna mempercepat proses pendinginan. Respon ini dikonfirmasi melalui grafik suhu yang menunjukkan penurunan suhu dari 30,37°C menuju sekitar 25°C secara signifikan. Setelah suhu mendekati nilai *setpoint*, sistem kendali menunjukkan respon *output* PWM yang cukup stabil tanpa fluktuasi berlebihan. Hal ini mencerminkan performa pengendalian yang baik dalam menjaga suhu tetap berada di sekitar nilai target.

Secara keseluruhan, kedua grafik ini menunjukkan bahwa sistem kendali PID telah berfungsi cukup baik dalam menjaga kestabilan suhu *Aquascape*. Sistem mampu merespons perubahan suhu dengan cepat dan mempertahankan suhu di sekitar nilai yang diharapkan.

Pengujian Sistem Terhadap Pertumbuhan *Java Moss*

Pengujian ini bertujuan untuk mengetahui pertumbuhan *Java Moss* selama empat minggu pada sistem *Aquascape* yang telah dikendalikan suhu airnya. Pengamatan dilakukan setiap minggu dengan mencatat panjang rata-rata *Java Moss* menggunakan mistar. Pengamatan dimulai sejak awal bulan April, dengan total empat kali pencatatan. Hasil yang diperoleh menunjukkan adanya peningkatan panjang secara bertahap dari minggu ke minggu, yang menunjukkan bahwa *Java Moss* dapat tumbuh dengan baik dalam kondisi lingkungan yang dikontrol dengan suhu 25°C dan pencahayaan selama 8 jam tiap hari.

Perhitungan rata-rata pertumbuhan tanaman:

- Sampel A:
 $(4.0 - 2.0)\text{cm} / 3 = 2.0 / 3 = 0.67 \text{ cm/minggu} \dots\dots\dots (20)$
- Sampel B:
 $(4.2 - 2.1)\text{cm} / 3 = 2.1 / 3 = 0.70 \text{ cm/minggu} \dots\dots\dots (21)$
- Sampel C:
 $(3.9 - 2.0)\text{cm} / 3 = 1.9 / 3 = 0.63 \text{ cm/minggu} \dots\dots\dots (22)$
- Sampel D:
 $(4.3 - 2.2)\text{cm} / 3 = 2.1 / 3 = 0.70 \text{ cm/minggu} \dots\dots\dots (23)$
- Sampel E:
 $(4.2 - 2.1)\text{cm} / 3 = 2.1 / 3 = 0.70 \text{ cm/minggu} \dots\dots\dots (24)$
- Sampel F:
 $(4.0 - 2.0)\text{cm} / 3 = 2.0 / 3 = 0.67 \text{ cm/minggu} \dots\dots\dots (25)$

Perhitungan total rata-rata pertumbuhan:

$$\frac{0,67+0,70+0,63+0,70+0,70+0,67}{6} = \frac{4,07}{6}$$

$$= 0,6783\text{cm/minggu} = 0,68\text{cm/minggu} \dots\dots\dots (26)$$

Tabel 7. Hasil Pengukuran Sampel

Tanggal	Sampel A	Sampel B	Sampel C	Sampel D	Sampel E	Sampel F
1/4/2025	2.0 cm	2.1 cm	2.0 cm	2.2 cm	2.1 cm	2.0 cm
8/4/2025	2.6 cm	2.8 cm	2.6 cm	2.9 cm	2.8 cm	2.6 cm
15/4/2025	3.3 cm	3.5 cm	3.2 cm	3.6 cm	3.5 cm	3.3 cm
22/4/2025	4.0 cm	4.2 cm	3.9 cm	4.3 cm	4.2 cm	4.0 cm
Rata-rata sample	0.67 cm	0.70 cm	0.63 cm	0.70 cm	0.70 cm	0.67 cm

Tabel 8. Perbandingan Visual

Minggu ke-	Dokumentasi	Keterangan
1		Awal penanaman <i>Java Moss</i> pada batok kelapa dengan panjang diameter 3cm.
2		Pada minggu berikutnya <i>Java Moss</i> mulai tumbuh secara memanjang.
3		Pada minggu ketiga <i>Java Moss</i> tumbuh lebih panjang dan lebih lebat daripada minggu sebelumnya
4		Pada minggu ke empat atau sudah genap satu bulan <i>Java Moss</i> terlihat lebih lebat dan rimbun daripada minggu ke 3

Selama periode pengamatan tiga minggu, setiap sampel *Java Moss* menunjukkan tingkat pertumbuhan yang cukup

stabil. Sampel A dan Sampel F mengalami pertumbuhan sebesar 2,0 cm, yang menghasilkan rata-rata pertumbuhan mingguan sebesar sekitar 0,67 cm. Sampel B, Sampel D, dan Sampel E menunjukkan pertumbuhan paling tinggi dengan rata-rata masing-masing sekitar 0,70 cm per minggu, mengindikasikan bahwa ketiga sampel tersebut merespons lingkungan tumbuh secara optimal.

Sementara itu, Sampel C menunjukkan pertumbuhan paling rendah dibandingkan sampel lainnya, dengan rata-rata pertumbuhan sebesar sekitar 0,63 cm per minggu. Meskipun selisihnya tidak besar, hal ini mungkin mencerminkan adanya perbedaan kecil dalam kondisi mikro atau karakteristik individu dari tanaman.

Secara keseluruhan, rata-rata pertumbuhan dari semua sampel adalah sekitar 0,68 cm per minggu. Nilai ini menunjukkan bahwa *Java Moss* dapat tumbuh secara konsisten dan sehat dalam kondisi yang digunakan selama percobaan. Variasi kecil antar sampel menunjukkan bahwa meskipun kondisi tumbuh cukup merata, faktor internal tanaman tetap mempengaruhi laju pertumbuhan.

PENUTUP

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. Penelitian ini berhasil menghasilkan sistem automasi lampu dan pendingin suhu berbasis *Peltier* yang dikendalikan algoritma PID serta terintegrasi dengan IoT. Seluruh komponen sensor DS18B20, modul *Peltier* dan *waterblock*, driver MOSFET, ESP32, dan lampu LED berfungsi secara terintegrasi sehingga pemantauan serta pengaturan suhu dan pencahayaan dapat diakses dan dikonfigurasi jarak jauh melalui jaringan pada platform BLYNK.
2. Hasil pengujian menunjukkan parameter akhir PID ($K_p=125$, $K_i=1$, $K_d=700$) mampu menurunkan dan menjaga suhu *Aquascape* pada *set-point* 25 °C dengan *overshoot* (1,48%), error *steady-state* 0 °C, *rise-time* 429s, serta *settling-time* 7923s. Hal ini menegaskan bahwa kendali PID memberikan stabilitas termal yang memadai untuk aplikasi pendinginan berbasis *Peltier*.
3. Integrasi pengaturan lampu berbasis NTP menyala otomatis 8 jam per hari bersama kontrol suhu stabil pada 25 °C menghasilkan lingkungan tumbuh yang konsisten. Rerata laju pertumbuhan *Java Moss* tercatat 0,68 cm per minggu dengan variasi antar sampel kecil, mengindikasikan kondisi suhu dan pencahayaan yang telah dioptimalkan mendukung kesehatan tanaman.

Saran

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya.

- Penambahan *peltier* untuk mencapai suhu kompensasi panas dan dingin lebih cepat dan mencapai *setpoint* lebih cepat.
- Penambahan *peltier* pada sisi *waterblock* untuk menghangatkan *waterblock* supaya ada kendali suhu panas.
- Penambahan kendali pompa untuk pemberian pupuk nutrisi *Java Moss*.

DAFTAR PUSTAKA

- Adi, W. (2019). Sistem kendali dan monitoring peralatan elektronik berbasis NodeMCU ESP8266 dan aplikasi Blynk. *Electrans*, 4(3), 3–11.
- Agus Slamet Wahyudi, Muhammad Rifa'i, & Tundung Subali Patma. (2024). Sistem pendinginan suhu menggunakan metode PID pada aquascape berbasis Arduino. *Jurnal Multidisiplin Saintek*, 3(12).
- Akbar Sujiwa, M. A. N. (2019). Analisa suhu pendinginan termoelektrik tipe TEC-12706 dengan variasi metode pendinginan sisi panas. *Seminar Nasional FORTEL*, 7(2), 1–6.
- Andi Setiadi. (2015, February 15). Cara menurunkan suhu aquascape. <https://indoaquascape.com/cara-menurunkan-suhu-aquascape/>
- Bella Puji Darma Samodra. (2019). Studi analisis penggunaan heatsink tembaga terhadap jumlah air tawar yang dihasilkan pada atmospheric water generator menggunakan thermoelectric cooler untuk kebutuhan air minum di lifeboat [Skripsi, Institut Teknologi Sepuluh Nopember]. <https://repository.its.ac.id/69326/>
- Bizzy, I. (2018). Modul termoelektrik (Teori & Aplikasi). Noer Fikri Offset.
- ESP32 Series Datasheet Version 4.7 2.4 GHz Wi-Fi + Bluetooth ® + Bluetooth LE SoC Including. (n.d.). Espressif Systems. <https://www.espressif.com>
- Goffinet, B., & Shaw, A. J. (2008). *Bryophyte biology* (2nd ed.). Cambridge University Press. <https://books.google.co.id/books?id=te0fAwAAQBAJ>
- Gunawan, I., Akbar, T., & Ilham, M. G. (2020). Prototipe penerapan Internet of Things (IoT) pada monitoring level air tandon menggunakan NodeMCU ESP8266 dan Blynk. *Jurnal Informatika dan Teknologi*, 3(1), 1–7.
- Hariyatno, H., Isanawikrama, I., Wimpertiwi, D., & Kurniawan, Y. J. (2018). Membaca peluang merakit 'uang' dari hobi aquascape. *Jurnal Pengabdian dan Kewirausahaan*, 2(2). <https://doi.org/10.30813/jpk.v2i2.1364>

Sistem Pengkondisian Suhu Berbasis *Peltier* Menggunakan Kendali PID Integrasi IoT Optimalisasi Pertumbuhan
Java Moss Aquascape

- Hercog, D., Lerher, T., Truntiĉ, M., & TeĹak, O. (2023). Design and implementation of ESP32-based IoT devices. *Sensors*, 23(15), 1–20. <https://doi.org/10.3390/s23156739>
- Hosokawa, S., Nakamura, Y., & Kuwae, T. (2009). Increasing temperature induces shorter leaf life span in an aquatic plant. *Oikos*, 118(8), 1158–1163. <https://doi.org/10.1111/j.1600-0706.2009.17288.x>
- Jose, A., & Joseph, D. E. (2022). A study of heavy metal bioaccumulation effects on chlorophyll content of the two aquatic bryophytes *Taxiphyllum barbieri* (Java Moss) and *Vesicularia montagnei* (Christmas Moss).
- Khoerun, B., Aziz, R., Sugara, F., Apriyanto, K., Putri, N., & Ramdhan, A. (2023). Kontrol sistem pendingin temperatur air dan larutan nutrisi hidroponik selada. *Jurnal Teknologi dan Sains*, 2(2).
- Kumar, N. V. (2021). Art and science of aquascaping. *The Pharma Journal*. <http://www.thepharmajournal.com>
- Muhamad Putra Nur Ismail, & Rosyani, P. (2024). Rancang bangun sistem automasi monitoring suhu dan lampu aquascape berbasis IoT (Internet of Things). *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, 2(5). <https://journal.mediapublikasi.id/index.php/logic>
- Muhibburrahman, I. (2022). Rancang bangun prototype pencampur cat duco otomatis berbasis Arduino Mega [Laporan proyek].
- NAZIF, & Muhammad Naufal. (2018). Pengaturan suhu pembuatan garam menggunakan DS18B20 serta monitoring menggunakan HMI pada proses pembuatan garam berbasis programmable logic controller (PLC). http://eprints.undip.ac.id/67144/6/Bab_2.pdf
- Natasya, Y., & Santoso, H. (2023). Prototipe aplikasi smart lighting untuk mengontrol lampu jalan berbasis Android menggunakan ESP32. *SIBATIK Journal*, 2(8). <https://doi.org/10.54443/sibatik.v2i8.1298>
- Ogata, K. (2010). *Modern control engineering* (5th ed.). Prentice-Hall.
- Ony Asrarul Qudsi, et al. (2019). Desain dan implementasi pengaturan kecepatan motor BLDC melalui pengaturan fluks. *Inovtek Seri Elektro*, 1(1), 20–28. <https://ejournal.polbeng.ac.id/index.php/ISE/article/view/1231/pdf>
- Pramadana, M. H., Rivai, M., & Pirngadi, H. (2021). Sistem kontrol pencahayaan matahari pada aquascape. *Jurnal Teknik ITS*, 10(1), 15–21. <https://doi.org/10.12962/j23373539.v10i1.59809>
- Radouan Ait Mouha. (2021). Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, 9(2), 77–101. <https://doi.org/10.4236/jdaip.2021.92006>
- Raharjo, S., Kurniawan, E., & Nurcahya, E. D. (2018). Sistem otomatisasi fotosintesis buatan pada aquascape berbasis Arduino. *Komputek*, 2(1), 39–45. <https://doi.org/10.24269/jkt.v2i1.66>
- Rusimamto, P. W., et al. (2021). Implementation of Arduino Pro Mini and ESP32-CAM for temperature monitoring on automatic thermogun IoT-based. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(3), 1366–1375. <https://doi.org/10.11591/ijeecs.v23.i3.pp1366-1375>
- Setya Wijaya, E., et al. (2021). Penerapan logika Fuzzy Tsukamoto untuk pemantauan kestabilan suhu menggunakan sensor DS18B20 pada styrofoam box pengemasan ikan. *Journal of Science and Technology Naskah*.
- Viktor Ardy Setiawan Ramby. (2023). Teknologi tepat guna sistem power storage pembangkit listrik tenaga surya untuk irigasi pertanian di Desa Kalijaran [Laporan tugas akhir, Politeknik Negeri Cilacap]. [https://elib.pnc.ac.id/917/3/BAB II.pdf](https://elib.pnc.ac.id/917/3/BAB%20II.pdf)
- Weisrawei, Y., Arman Prasetya, D., & Boedi Setiawan, A. (2018). Perancangan smart green house dengan optimalisasi pH dan suhu air pada tanaman selada, media tanam hidroponik berbasis Arduino Uno. *SinarFe*, 7(1).
- Zhang, P., et al. (2020). Interactive effects of rising temperature and nutrient enrichment on aquatic plant growth, stoichiometry, and palatability. *Frontiers in Plant Science*, 11, Article 58. <https://doi.org/10.3389/fpls.2020.00058>