

## SISTEM KONTROL LAMPU *AQUASCAPE* MENGGUNAKAN *RULE BASED SYSTEM* DAN PID

Primus Mario Andaka Ginting<sup>1</sup>, Suhardi<sup>2</sup>, Rahmi Hidayati<sup>3</sup>

<sup>1,2,3</sup> Rekayasa Sistem Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura

<sup>1</sup>H1051211073@student.untan.ac.id, <sup>2</sup> suhardi@siskom.untan.ac.id, <sup>3</sup> rahmihidayati@siskom.untan.ac.id

### Abstrak

Energi listrik berperan penting dalam kehidupan, namun konsumsi yang tidak efisien, khususnya pada pencahayaan, dapat membebani biaya dan sumber daya. Pada ekosistem buatan seperti *aquascape*, pencahayaan tidak hanya berfungsi sebagai estetika tetapi juga krusial bagi kesehatan biota di dalamnya, karena intensitas yang tidak tepat dapat memicu tekanan pada fauna dan pertumbuhan alga yang tidak diinginkan. Penelitian ini bertujuan mengembangkan sistem kontrol lampu otomatis hemat energi untuk *aquascape* menggunakan mikrokontroler NodeMCU ESP32 dan sensor LDR. Sistem ini mengintegrasikan logika *Rule Based System* untuk adaptasi terhadap kondisi cahaya lingkungan dengan kontroler *Proportional Integral Derivative* (PID) untuk optimasi stabilitas dan kecepatan respons. Parameter PID ditentukan menggunakan dua metode, yaitu *Ziegler Nichols Open Loop* dan *Closed Loop*, untuk kemudian dibandingkan performa dari kedua metode tersebut. Hasil perbandingan menunjukkan bahwa metode *Ziegler Nichols Closed Loop* memperoleh nilai  $K_p = 0,72$ ,  $K_i = 0,48$ , dan  $K_d = 0,28$  sehingga memberikan kinerja yang jauh lebih optimal dibandingkan metode *Open Loop* yang nilainya relatif kecil dengan nilai  $K_p = 0,0525$ ,  $K_i = 0,00328$ , dan  $K_d = 0,21$ . Sistem yang dihasilkan dari metode *Closed Loop* terbukti lebih cepat mencapai target dan mampu mengoreksi *error* secara cepat, sehingga berhasil menciptakan sistem pencahayaan yang adaptif dan stabil.

**Kata kunci:** Lampu, *Rule Based System*, PID, *Aquascape*, LDR

### 1. Pendahuluan

Energi listrik merupakan kebutuhan penting dalam kehidupan sehari-hari. Penggunaan listrik yang kurang terkontrol dapat menyebabkan pemborosan, salah satunya adalah penggunaan lampu yang menyala terus menerus meskipun tidak diperlukan, misalnya pada siang hari (Muhazir & Sihombing, 2021). Perkembangan teknologi lampu, seperti LED, menawarkan kelebihan dari segi penggunaan daya yang lebih rendah dan umur pakai yang lebih lama dibandingkan lampu konvensional, yang sesuai dengan upaya penghematan listrik (Asmaleni et al., 2020).

Pada *aquascape*, atau seni menata tanaman air dalam akuarium, pencahayaan memiliki fungsi yang krusial. Cahaya tidak hanya untuk estetika, tetapi juga untuk mendukung proses fotosintesis tanaman. Namun, intensitas cahaya yang tidak tepat, kekurangan maupun kelebihan cahaya, dapat mengganggu kesehatan fauna memicu perkembangan alga yang tidak diinginkan (Kusumo et al., 2022). Oleh karena itu, dibutuhkan sebuah sistem kontrol yang dapat mengatur pencahayaan secara otomatis.

Penelitian ini dibangun berdasarkan acuan dari penelitian sebelumnya yang relevan. Penelitian oleh (Aribowo et al., 2022) berhasil memanfaatkan sensor LDR untuk efisiensi energi pada lampu penerangan jalan, namun belum mengimplementasikan kontroler PID untuk optimasi respons. Dari sisi metodologi, penerapan *Rule Based System* merujuk pada penelitian oleh (Juanda & Yadi,

2020) yang menunjukkan efektivitas logika berbasis aturan, meskipun pada domain yang berbeda. Sementara itu, penelitian terdahulu oleh (Putra et al., 2024) juga merancang sistem automasi untuk *aquascape*, namun perbedaannya terletak pada penggunaan sensor suara sebagai pengendali, sedangkan penelitian ini berfokus pada sensor LDR yang adaptif terhadap cahaya lingkungan.

Penelitian ini bertujuan membangun sistem kontrol lampu otomatis untuk *aquascape* yang menggabungkan *Rule Based System* dengan kontroler *Proportional Integral Derivative* (PID). Sistem ini menggunakan NodeMCU ESP32 dan sensor LDR untuk membaca kondisi cahaya sekitar. Berdasarkan aturan *if-then*, sistem menentukan tingkat kecerahan lampu yang dibutuhkan, dan kontroler PID memastikan perubahan kecerahan tersebut berjalan dengan stabil dan cepat mencapai nilai yang dituju.

### 2. Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahapan, yaitu studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.

#### 2.1 Studi Literatur

Tahap ini melibatkan pengumpulan informasi dari berbagai sumber seperti jurnal penelitian sebelumnya, buku referensi, dan artikel ilmiah yang relevan dengan topik lampu LED, *aquascape*, *Rule*

*Based System*, dan kontroler PID. Studi literatur bertujuan untuk memahami dasar teori, mengidentifikasi penelitian terdahulu, dan menentukan kontribusi dari penelitian ini.

## 2.2 Pengumpulan Data

Data untuk penelitian ini diperoleh langsung dari sistem yang dirancang. Mikrokontroler NodeMCU ESP32 digunakan untuk membaca nilai analog dari sensor LDR secara *real-time*. Data berupa nilai intensitas cahaya kemudian menjadi masukan utama untuk diproses oleh logika *Rule Based System* dan algoritma kontrol PID guna menentukan keluaran yang sesuai.

## 2.3 Analisis Kebutuhan

Analisis kebutuhan sistem dibagi menjadi perangkat keras dan lunak. Perangkat keras yang digunakan meliputi:

1. NodeMCU ESP32 sebagai pusat kendali, komponen ini dipilih karena telah terintegrasi dengan fitur Wi-Fi dan *Bluetooth* yang sangat cocok untuk aplikasi *Internet of Things* (IoT) (Harpad et al., 2022).
2. Sensor LDR untuk mendeteksi intensitas cahaya, sensor LDR merupakan detektor cahaya yang andal di mana nilai hambatannya berubah secara signifikan berdasarkan intensitas cahaya yang diterima (Alamsyah et al., 2022).
3. Modul *Relay* untuk mengontrol daya lampu, *relay* membutuhkan daya yang sangat kecil untuk mengaktifkan kontaktornya, tetapi mampu mengontrol perangkat yang membutuhkan daya yang lebih besar (Alfariski et al., 2022).
4. Lampu LED sebagai *output* pencahayaan, LED dibuat menggunakan material semikonduktor. Jenis bahan semikonduktor yang digunakan akan menentukan warna cahaya yang dihasilkan oleh LED (Alferinanda, 2020).

## 2.4 Rule Based System

Sistem berbasis aturan (*Rule Based System*) tersusun dari aturan-aturan yang digunakan sistem untuk melakukan penalaran dari fakta menuju kesimpulan (Juanda & Yadi, 2020). Sistem ini menggunakan teknik dasar berupa aturan *if-then*, dalam aturan logika *if-then*, bagian *if* disebut *anteseden*, yaitu bagian yang memuat kondisi, fakta, atau pernyataan awal. Sedangkan bagian *then* disebut *konsekuen*, yaitu bagian yang menyatakan hasil atau kesimpulan yang diambil jika kondisi pada *anteseden* terpenuhi. Bentuk umum dari aturan ini dapat dituliskan sebagai:

*IF*<*anteseden*> *THEN*<*konsekuen*>

Fakta-fakta dalam bagian *anteseden* dianggap benar dan digunakan sebagai dasar pengambilan keputusan (Setiawan et al., 2020). Sebuah *anteseden*

dapat terdiri dari satu pernyataan tunggal atau beberapa pernyataan yang dihubungkan dengan operator logika seperti AND (dan) atau OR (atau). Penggunaan operator logika tersebut bertujuan untuk menggabungkan lebih dari satu kondisi dalam satu aturan (Abdillah Nurkholis, 2019). Dalam mengembangkan sistem pendukung keputusan berbasis aturan, sistem dirancang untuk menentukan solusi terkait (Rakhmawati et al., 2018).

## 2.5 Proportional Integral Derivative (PID)

Kontrol PID adalah salah satu metode pengendalian yang bekerja secara otomatis untuk menyesuaikan sistem dengan *set point* yang telah ditentukan. Kontroler ini memiliki tiga parameter, *Proportional* (P) untuk memberikan penguatan, *Integral* (I) untuk menghilangkan kesalahan tetap, dan *Derivative* (D) untuk memberikan redaman dan mengurangi lonjakan. PID berfungsi untuk mempercepat respon dari sistem dan mencegah *overshoot* (Tika et al., 2021). Sistem ini memungkinkan perancangan karakteristik respons transien sesuai kebutuhan. Secara matematis, PID *controller* dirumuskan sebagai:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Keterangan:

$u(t)$  = sinyal control

$e(t)$  = *error*

$K_p$  = konstanta proporsional

$K_i$  = konstanta Integral

$K_d$  = konstanta derivatif

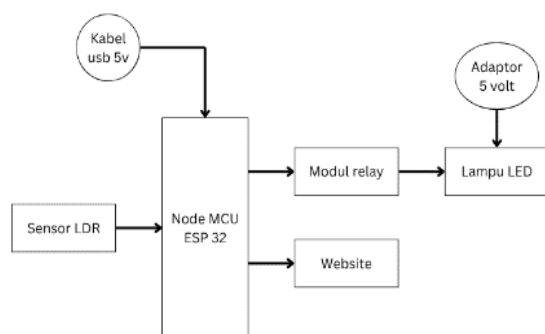
Pecahan dari persamaan umum PID ditentukan sesuai dengan masing-masing konstanta, *set point* harus ditentukan untuk mengukur atau membuat sistem sesuai dengan yang diinginkan, kemudian konstanta proporsional, Integral, dan derivatif membuat sistem yang dirancang akan memberikan respon yang cepat terhadap *error*, presisi yang tinggi dan mengurangi *overshoot* (Agustian et al., 2022).

Penentuan parameter PID dapat dilakukan dengan berbagai metode, salah satunya adalah *Ziegler Nichols* yang terbukti efektif dalam berbagai implementasi kontrol (Agustian et al., 2022).

## 3. Perancangan dan Implementasi

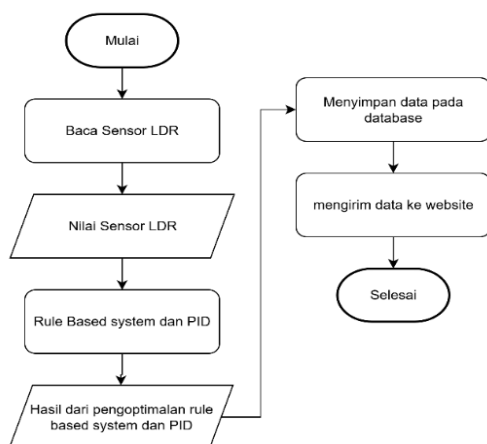
### 3.1 Perancangan Sistem

Perancangan sistem mencakup proses penyambungan seluruh komponen perangkat keras serta pengembangan perangkat lunak. Gambar 1 merupakan diagram sistem kontrol lampu *aquascape*.



Gambar 1. Perancangan Arsitektur Sistem

Perancangan arsitektur sistem menggunakan NodeMCU ESP32 sebagai mikrokontroler, kemudian kabel usb 5v yang dicolokkan pada laptop sebagai sumber daya dan sebagai penerima data dari ESP32, sensor LDR terkoneksi pada ESP32 yang berfungsi untuk menangkap tegangan analog dan merubahnya menjadi digital, modul *relay* yang terkoneksi dengan ESP32 berfungsi untuk membatasi dan memutuskan aliran lampu ketika intensitas cahaya yang diterima besar, lampu LED sebagai cahaya atau hasil dari perancangan sistem yang telah dibuat. LED membutuhkan adaptor 5v untuk menyalakan lampu tersebut. Pada perancangan perangkat lunak kode program ditulis pada Arduino IDE agar NodeMCU ESP32 dapat berjalan. Gambar 2 merupakan diagram alir sistem kontrol lampu *aquascape*.

Gambar 2. Diagram Alir Sistem Kontrol Lampu *Aquascape*

NodeMCU ESP32 merupakan mikrokontroler yang digunakan pada penelitian ini, NodeMCU ESP32 juga berfungsi sebagai otak dari sistem yang dibuat, setelah kode program dijalankan NodeMCU ESP32 akan membaca nilai dari sensor LDR, kemudian setelah nilai tersebut ditangkap maka *Rule Based System* akan memberikan aturan yang sesuai dengan empat kondisi pada aturan yang sudah ditentukan, setelah itu PID akan memberikan respon cepat terhadap sistem agar lampu bisa mencapai *setpoint* dengan tepat dan tidak *overshoot*, PID juga menghitung *error* yang terjadi pada sistem, hal ini membuat sistem akan mengatasi *error* secara cepat dan sistem menjadi responsif, setelah itu NodeMCU

akan mengirimkan data ke *database* kemudian dari *database* akan mengirim ke *website*.

### 3.2 Perancangan Rule Based System

Perancangan *Rule Based System* dalam penelitian ini bertujuan untuk membuat logika kontrol atau *setpoint* bagi lampu. Sistem ini menggunakan aturan logika *if-then* untuk menentukan tingkat kecerahan lampu yang diinginkan berdasarkan nilai yang dibaca oleh sensor LDR. Pendekatan yang digunakan adalah *forward chaining*, di mana data masukan dari sensor (fakta) diproses untuk menghasilkan kesimpulan berupa nilai keluaran untuk lampu. Sensor LDR menghasilkan nilai masukan antara 1 (sangat terang) hingga 4095 (sangat gelap). Sebaliknya, nilai keluaran untuk lampu menggunakan sinyal *Pulse Width Modulation* (PWM) dengan rentang 0 (mati) hingga 255 (terang maksimal). Nilai-nilai *setpoint* PWM yang akan diimplementasikan pada sistem ini adalah 0, 32, 64, dan 128. Nilai-nilai ini didasarkan pada *duty cycle* 0%, 12,5%, 25%, dan 50% dari resolusi PWM 8-bit (255). Untuk mengimplementasikan logika ini, rentang nilai LDR dibagi menjadi empat kondisi. Setiap kondisi akan menentukan nilai *setpoint* yang harus dicapai oleh lampu,

Nilai PWM pada sistem ini menggunakan resolusi 8-bit (255), yang berarti bahwa setiap nilai kecepatan direpresentasikan dengan rentang angka dari 0 hingga 255. Adapun nilai-nilai PWM yang akan diimplementasikan pada sistem adalah sebagai berikut:

*Duty cycle* 0%

$$PWM = \text{Duty cycle} \times \text{PWM resolution value}$$

$$PWM = 0\% \times 255$$

$$PWM = 0$$

Ketika *duty cycle* bernilai 0% dan resolusi yang digunakan adalah 8 bit, maka nilai *duty cycle* direpresentasikan dengan angka 0 hingga 255 untuk nilai PWM sebesar 0.

*Duty cycle* 12,5%

$$PWM = \text{Duty cycle} \times \text{PWM resolution value}$$

$$PWM = 12,5\% \times 255$$

$$PWM = 32$$

Ketika *duty cycle* bernilai 12,5% dan resolusi yang digunakan adalah 8 bit, maka nilai *duty cycle* direpresentasikan dengan angka 0 hingga 255 untuk nilai PWM sebesar 32.

*Duty cycle* 25%

$$PWM = \text{Duty cycle} \times \text{PWM resolution value}$$

$$PWM = 25\% \times 255$$

$$PWM = 64$$

Ketika *duty cycle* bernilai 25% dan resolusi yang digunakan adalah 8 bit, maka nilai *duty cycle* direpresentasikan dengan angka 0 hingga 255 untuk nilai PWM sebesar 64.

*Duty cycle* 50%

$$PWM = \text{Duty cycle} \times \text{PWM resolution value}$$

$$PWM = 50\% \times 255$$

PWM = 128

Ketika *duty cycle* bernilai 50% dan resolusi yang digunakan adalah 8 bit, maka nilai *duty cycle* direpresentasikan dengan angka 0 hingga 255 untuk menghasilkan nilai PWM sebesar 128. Tabel logika *Rule Based System* dapat dilihat pada Tabel 1.

Tabel 1. Logika *Rule Based System*

Kondisi	IF	THEN
	LDR value	Setpoint pada lampu
1	1-2500	0
2	2501-3000	32
3	3001-3500	64
4	3501-4095	128

### 3.3 Perancangan Ziegler Nichols Open Loop

Metode *Ziegler Nichols* merupakan metode yang menentukan konstanta *Proportional*, konstanta *Integral*, dan konstanta *Derivative* dengan menggunakan parameter yang dilanjutkan dengan perhitungan, tahapan pertama yaitu mencari nilai L (*dead time*) dan T (waktu pemulihan), L dan T merupakan nilai yang bertujuan untuk menghitung kecepatan dan ketepatan sistem. Nilai L dan T dapat dilihat pada Tabel 2.

Tabel 2. Nilai L (*dead time*) dan T (waktu pemulihan)

L	T
8 detik	0,35 detik

$$K_p = 1,2 \times \frac{0,35}{8} = 0.0525$$

$$T_i = 2 \times L = 2 \times 8 = 16 \text{ detik}$$

$$T_d = 0,5 \times 8 = 4 \text{ detik}$$

Setelah melakukan perhitungan pada parameter, nilai  $T_i$  dan  $T_d$  telah ditentukan yang dimana nilai tersebut dilakukan untuk perhitungan lanjutan untuk menentukan nilai  $K_p$ ,  $K_i$ , dan  $K_d$ .

$$K_p = 0.0525$$

$$K_i = \frac{K_p}{T_i} = \frac{0,0525}{16} = 0.00328$$

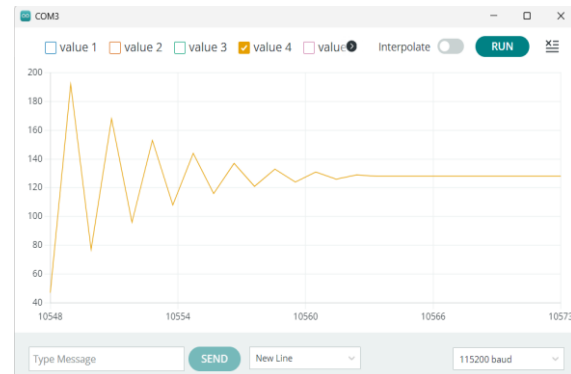
$$K_d = K_p \times T_d = 0,0525 \times 4 = 0,21$$

Nilai  $K_p$ ,  $K_i$ , dan  $K_d$  telah ditentukan yaitu nilai  $K_p = 0,0525$ ,  $K_i = 0,00328$  dan  $K_d = 0,21$ .

### 3.4 Perancangan Ziegler Nichols Closed Loop

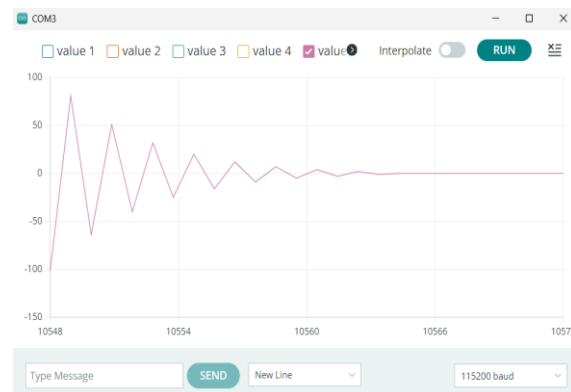
*Closed Loop* merupakan metode yang menonaktifkan *Integral* dan *Derivative* pada pengendali, yang kemudian menaikkan nilai *Proportional* hingga mencapai osilasi yang terus berlanjut. Ketika osilasi terjadi maka terjadi periode waktu antara 2 puncak osilasi yang disimbolkan dengan  $P_u$  atau *ultimate period*.

Pada  $K_p = 1,2$  sistem mendapatkan lonjakan pada detik pertama yang dimana ini yang disebut osilasi, terjadi karena nilai  $K_p$  sangat besar dan sistem akan mengalami *error* dan tidak stabil. Sistem yang menggunakan  $K_p = 1,2$  dapat dilihat pada Gambar 3.



Gambar 3. Hasil Penggunaan Nilai  $K_p = 1,2$  (Cahaya)

Gambar 3 adalah visualisasi dari *brightness* yang merepresentasikan *output* aktual dari sistem, yaitu tingkat kecerahan. Grafik tersebut adalah cerminan dan kinerja sistem kontrol. Fase kenaikan yang cepat menunjukkan bahwa sistem sedang aktif bekerja, menyesuaikan *output* untuk mengejar nilai target. Fase di mana grafik menjadi datar menunjukkan bahwa sistem telah mencapai kondisi target atau *steady state*. Artinya, sistem telah berhasil mencapai tujuannya.



Gambar 4. Hasil Penggunaan Nilai  $K_p = 1,2$  (Error)

Gambar 4 merupakan analisis sistem kontrol dan nilai *error*, nilai ini dihitung dari selisih antara nilai target dan nilai aktual yang terukur. Sistem masih mengoreksi *error* secara lambat atau kurang cepat. Pada grafik tidak ditampilkan waktu untuk mencari nilai  $P_u$ , maka waktu diambil dari *serial monitor*, hasil dari pengambilan *brightness*, *error*, dan waktu dapat dilihat pada Tabel 3.

Tabel 3. Hasil Nilai *Brightness*, *Error*, dan waktu

<i>Brightness</i>	<i>Error</i>	Waktu
190	70	988 ms
80	-40	1988 ms
170	50	2988 ms

Jika telah mendapatkan nilai yang beresilasi maka langkah selanjutnya adalah menentukan nilai  $K_u$  dan  $P_u$  untuk melanjutkan perhitungan *Closed Loop*, tabel nilai  $K_u$  dan  $P_u$  dapat dilihat pada Tabel 4.

Tabel 4. Tabel nilai  $K_u$  dan  $P_u$

$K_u$	$P_u$
1,2	3 detik

Setelah mendapatkan nilai  $K_u$  dan  $P_u$  langkah berikutnya yang dilakukan adalah perhitungan dengan metode *Ziegler Nichols Closed Loop*.

$$K_p = 0,6 \times K_u$$

$$T_i = 0,5 \times P_u$$

$$T_d = 0,13 \times P_u$$

$$K_p = 0,6 \times 1,2 = 0,72$$

$$T_i = 0,5 \times 3 = 1,5 \text{ detik}$$

$$T_d = 0,13 \times 3 = 0,39 \text{ detik}$$

$$K_i = \frac{K_p}{T_i} = \frac{0,72}{1,5} = 0,48$$

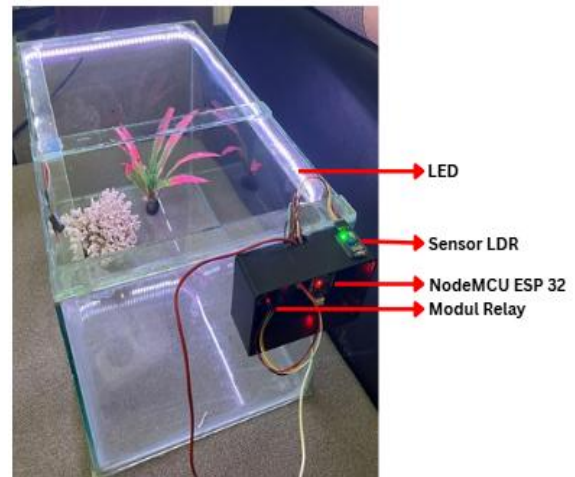
$$K_d = K_p \times T_d = 0,72 \times 0,39 = 0,2808$$

Nilai  $K_p$ ,  $K_i$  dan  $K_d$  pada metode *Closed Loop* telah ditentukan yaitu  $K_p = 0,72$ ,  $K_i = 0,48$ , dan  $K_d = 0,28$ , setelah nilai didapat maka selanjutnya melakukan pengujian pada sistem dengan metode tersebut.

#### 4. Hasil dan Pembahasan

Pada implementasi perangkat keras melakukan penyusunan komponen perangkat keras yang dirancang sebelumnya. Mikrokontroler yang menjadi komponen utama yaitu NodeMCU ESP32. Sistem ini memanfaatkan modul NodeMCU ESP32 yang terintegrasi dengan sensor LDR untuk melakukan penyesuaian otomatis terhadap intensitas cahaya lingkungan. Sebelum proses implementasi, dilakukan kalibrasi sensor dengan menetapkan nilai ambang pada kondisi gelap dan terang guna memastikan akurasi pembacaan. Apabila nilai LDR berada di luar rentang yang telah ditetapkan, sistem akan mengklasifikasikannya sebagai data tidak valid dan mempertahankan nilai terakhir yang terverifikasi untuk menjaga kestabilan operasi.

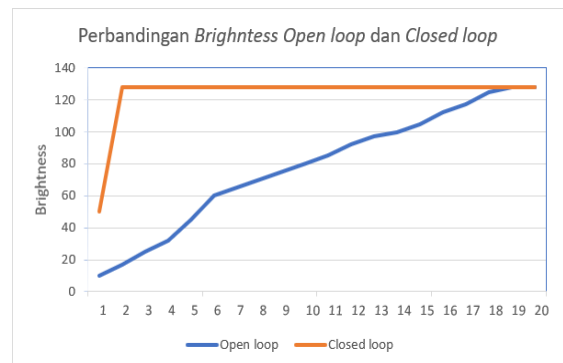
Selanjutnya, ketika terdapat penambahan atau perubahan aturan (*rule*), sistem dapat disesuaikan melalui pembaruan logika pada NodeMCU ESP32 agar tetap responsif dan adaptif terhadap dinamika kondisi lingkungan. Modul *relay* digunakan untuk mengurangi, memutuskan dan menjalankan arus yang masuk ke dalam lampu LED. Gambar 5 merupakan implementasi keseluruhan sistem.



Gambar 5. Implementasi Keseluruhan Sistem

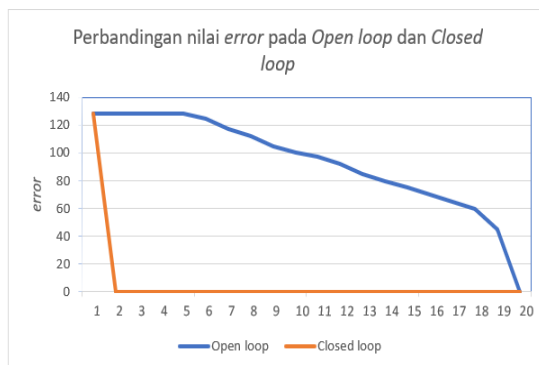
#### 4.1 Perbandingan Metode *Open Loop* dan *Closed Loop*

Perbedaan pada metode *Ziegler Nichols Open Loop* dan *Closed loop* terdapat pada *output* yang dikeluarkan sistem yaitu *brightness* dan *error*, nilai dari *brightness* yaitu *output* dari lampu, pada metode *Open Loop*, lampu belum menyala sesuai dengan target yang diinginkan, sementara *Closed Loop* sudah mencapai target dengan catatan waktu yang lebih singkat, hal ini membuktikan bahwa *Closed Loop* lebih optimal dan responsif pada sistem. Grafik perbandingan *brightness* dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan *Brightness Closed Loop* dan *Open Loop*

Nilai *error* didapat dari target *setpoint* dikurang *brightness*, hal ini juga menjadi pembeda karena data pada *Open Loop* kurang mampu mengatasi *error* dengan cepat, sementara *Closed Loop* mampu mengatasi *error* dengan sangat baik. Grafik perbandingan *error* dapat dilihat pada Gambar 7.



Gambar 7. Perbandingan Error Closed Loop dan Open Loop

#### 4.2 Pengujian Keseluruhan Sistem dengan Closed Loop

Pada pengujian keseluruhan sistem digunakan metode *Closed Loop* karena lebih optimal dibandingkan metode *Open Loop*, dengan data yang diambil selama satu hari penuh dan *output* setiap detik. Pengambilan data dibagi menjadi empat waktu: pagi hari, siang hari, sore hari, dan malam hari.

Pada pagi hari, intensitas cahaya terpantau berubah-ubah. Meskipun demikian, pengoreksian oleh kontroler PID sudah cukup bagus karena nilai tetap stabil setelah beberapa detik sistem mencapai target. Pada siang hari data sangat fluktuatif atau naik turun karena perubahan paparan sinar matahari dan awan. Namun, pengoreksian PID tetap bekerja dengan baik sehingga nilai menjadi stabil. Sore Hari Intensitas cahaya yang masuk pada sore hari cenderung kecil, sehingga *setpoint* berada pada nilai 64 dan 128. Perubahan yang terjadi tidak terlalu banyak jika dibandingkan dengan data pada siang hari. Dan malam Hari tidak ada perubahan intensitas cahaya yang terjadi. Akibatnya, *setpoint* sistem stabil pada nilai 128 dari awal hingga akhir pengambilan data.

Hasil dari pengujian yang telah dilakukan menunjukkan bahwa sistem yang dirancang dengan sensor LDR dan logika *Rule Based System* dapat bekerja, di mana sistem merespons perubahan cahaya secara *real-time* dan melakukan tindakan pengendalian sesuai aturan-aturan logika yang telah dirancang.

Meskipun *Rule Based System* bekerja dengan baik, pengujian memperlihatkan bahwa metode *Ziegler Nichols Open Loop* kurang sesuai untuk mendapatkan nilai konstanta kontrol  $K_p$ ,  $K_i$ , dan  $K_d$ . Nilai parameter yang diperoleh dari metode *Open Loop*, yaitu  $K_p = 0,0525$ ,  $K_i = 0,00328$ , dan  $K_d = 0,21$ , menghasilkan sistem yang kurang responsif terhadap perubahan yang terjadi. Hal ini disebabkan karena nilai konstanta yang didapat terlalu kecil, sehingga sistem kontrol menjadi lambat dalam memberikan respons. Akibatnya, sistem yang dirancang tidak dapat bekerja sesuai harapan awal, yaitu menghasilkan kontrol yang cepat dan stabil terhadap kondisi pencahayaan yang berubah-ubah.

Sebaliknya, penggunaan metode *Ziegler Nichols Closed Loop* memberikan hasil yang jauh lebih baik. Pada metode ini, pendekatan dilakukan dengan menaikkan nilai  $K_p$  secara bertahap sambil menonaktifkan  $K_i$  dan  $K_d$  hingga sistem menunjukkan osilasi berkelanjutan. Dari proses tersebut, diperoleh nilai  $K_u$  (*ultimate gain*) dan  $P_u$  (*ultimate period*) yang digunakan untuk menghitung konstanta kontrol. Hasil perhitungan memberikan nilai  $K_p = 0,72$ ,  $K_i = 0,48$ , dan  $K_d = 0,28$ , yang kemudian diimplementasikan pada sistem. Nilai-nilai tersebut terbukti menghasilkan cara kerja sistem yang jauh lebih baik dibandingkan dengan metode *Open Loop*, di mana sistem mampu merespons secara cepat dan stabil terhadap berbagai variasi pencahayaan.

Pengujian sistem selanjutnya dilakukan dengan melibatkan *Rule Based System* yang telah dirancang, di mana aturan logika diformulasikan berdasarkan tingkat intensitas cahaya pada pagi, siang, sore, dan malam hari. Total data yang diambil selama satu hari pengujian adalah sebanyak 1141 data, yang dianalisis untuk mengevaluasi cara kerja sistem kontrol otomatis yang telah dibuat.

Dengan demikian, dapat dinyatakan bahwa gabungan antara sensor LDR, *Rule Based System*, dan metode *tuning PID Ziegler Nichols Closed Loop* menunjukkan hasil kerja yang sangat baik terhadap sistem yang dirancang.

#### 5. Kesimpulan

Penelitian yang dilakukan dengan logika *Rule Based System* dan pendekatan PID bekerja dengan optimal, penerapan sistem dengan *Rule Based System* dan PID menunjukkan bahwa logika aturan dapat bekerja pada sistem dan PID berfungsi untuk menghitung *error* yang terjadi dan membuat respon sistem menjadi lebih cepat, serta PID mencegah osilasi dari sistem yang dirancang. Penelitian terkait sistem *control* untuk lampu hemat listrik dengan *Rule Based System* dan PID menunjukkan bahwa metode *Ziegler Nichols Open Loop* dan *Closed Loop* merupakan metode yang berfungsi untuk mencegah osilasi, mencegah *overshoot* dan juga menghitung *error* pada sistem, akan tetapi pada sistem yang dirancang dengan metode *Open Loop* dilakukan dengan pencarian nilai L dan T kemudian dilakukan perhitungan dengan parameter *Open Loop*.

Hasil yang didapat kurang optimal karena sistem terlalu lambat untuk mencapai target dan kurang responsif, nilai dari metode *Ziegler Nichols Open Loop* yaitu  $K_p = 0,0525$ ,  $K_i = 0,00328$ ,  $K_d = 0,21$ . Sementara *Closed Loop* lebih optimal dibanding *Open Loop* karena mendapatkan nilai  $K_p$  dengan cara menaikkan nilai tersebut sampai terjadi osilasi, kemudian setelah terjadi osilasi kita mencari nilai  $K_u$  dan  $P_u$ , kemudian menghitung nilainya sesuai dengan parameter *Closed Loop*, nilai yang optimal

dengan metode *Closed Loop* adalah  $K_p = 0,72$ ,  $K_i = 0,48$ , dan  $K_d = 0,28$ .

Penelitian selanjutnya dapat dikembangkan dengan menambahkan sensor suhu air untuk menjaga kondisi lingkungan *aquascape* lebih optimal. Selain itu, sistem dapat diintegrasikan dengan aplikasi berbasis IoT agar pengguna dapat memantau dan mengatur pencahayaan secara daring. Implementasi algoritma kontrol adaptif atau fuzzy PID juga dapat menjadi alternatif untuk meningkatkan presisi sistem terhadap perubahan intensitas cahaya yang lebih dinamis.

#### Daftar Pustaka:

- Abdillah Nurkholis, M. (2019). *Perancangan Aturan Kebijakan Ruang Berdasarkan Konteks Aktivitas Dan Jumlah Orang Menggunakan Rule-Based Untuk Pengendalian Perangkat Listrik*.
- Agustian, I., Prakoso, D. S., Faurina, R., & Daratha, N. (2022). Sistem Kendali Suhu Mesin Tetas Telur Ayam Buras Menggunakan Kontroler PID dengan Metode Tuning Ziegler Nichols Open Loop Step Response. *Jurnal Amplifier Mei*, 12.
- Alamsyah, N., Rahmani, H. F., & Yeni. (2022). Lampu Otomatis Menggunakan Sensor Cahaya Berbasis Arduino Uno dengan Alat Sensor LDR. *Formosa Journal of Applied Sciences*, 1(5), 703–712. <https://doi.org/10.55927/fjas.v1i5.1444>
- Alfariski, M. R., Dhandi, M., & Kiswantonono, A. (2022). Automatic Transfer Switch (ATS) Using Arduino Uno, IoT-Based Relay and Monitoring. *JTECS: Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem Dan Komputer*, 2(1), 1. <https://doi.org/10.32503/jtecs.v2i1.2238>
- Alferinanda, Y. (2020). *Efisiensi Penggunaan Energi pada Lampu Penerangan Jalan Raya*.
- Aribowo, D., Priyogi, G., & Islam, S. (2022). *APLIKASI SENSOR LDR (LIGHT DEPENDENT RESISTOR) UNTUK EFISIENSI ENERGI PADA LAMPU PENERANGAN JALAN UMUM*.
- Asmaleni, P., Hamdani, D., & Sakti, I. (2020). PENGEMBANGAN SISTEM KONTROL KIPAS ANGIN DAN LAMPU OTOMATIS BERBASIS SAKLAR SUARA MENGGUNAKAN ARDUINO UNO. *Jurnal Kumparan Fisika*, 3(1), 59–66. <https://doi.org/10.33369/jkf.3.1.59-66>
- Harpad, B., Salmon, S., & Saputra, R. M. (2022). SISTEM MONITORING KUALITAS UDARA DI KAWASAN INDUSTRI DENGAN NODEMCU ESP32 BERBASIS IOT. *Jurnal Informatika Wicida*, 12(2), 39–47. <https://doi.org/10.46984/inf-wcd.1955>
- Juanda, R., & Yadi, I. Z. (2020). Penerapan Rule Based Dengan Algoritma Viterbi Untuk Deteksi Kesalahan Huruf Kapital Pada Karya Ilmiah. In *Journal of Computer and Information Systems Ampera* (Vol. 1, Issue 1). <https://journalcomputing.org/index.php/journal-cisa/index>
- Kusumo, A. S., Rusimamto, P. W., Suprianto, B., Putu, G., & Buditjahjanto, A. (2022). *Sistem Kontrol Intensitas Cahaya Lampu Aquascape Menggunakan Fuzzy Logic Controller Berbasis Arduino*.
- Muhazir, A., & Sihombing, C. (2021). *SMART LED SEBAGAI SOLUSI LAMPU HEMAT LISTRIK*.
- Putra, M., Ismail, N., & Rosyani, P. (2024). *Rancang Bangun Sistem Automasi Monitoring Suhu Dan Lampu Aquascape Berbasis IoT (Internet of Things)*. <https://journal.mediapublikasi.id/index.php/logic>
- Rakhmawati, N. A., Budi, A. S., Altetiko, F. J., Ramadhani, F., Wardati, N. K., & Hindrayani, K. M. (2018). Penentuan Prioritas Pengambilan Pesanan Barang Oleh Angkutan Kota dengan Metode Rule-Based System. *JURNAL SISTEM INFORMASI BISNIS*, 8(2), 195. <https://doi.org/10.21456/vol8iss2pp195-202>
- Setiawan, D., Ishak, & Sembiring, M. A. (2020). Control system for adjusting the brightness level with PWM technique using visual net microcontroller-based. *Journal of Robotics and Control (JRC)*, 1(4), 102–108. <https://doi.org/10.18196/jrc.1422>
- Tika, S. W., Baqaruzi, S., & Muhtar, A. (2021). Perancangan Sistem Pemantauan dan Pengendalian Debit Air Menggunakan Kontrol PID. *Jurnal ELECTRON*, 2(1), 41–47.

*Halaman ini sengaja dikosongkan*